



UNIVERSIDAD CARLOS III

ESCUELA POLITÉCNICA SUPERIOR CARLOS III

Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

**Desarrollo de aplicaciones informáticas para
la gestión de información de una PyME**

Luis Marino López Luque

Septiembre/2015



UNIVERSIDAD CARLOS III
ESCUELA POLITÉCNICA SUPERIOR CARLOS III
Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

**Desarrollo de aplicaciones informáticas para
la gestión de información de una PyME**

Tutor/Director:
Autor:

Dolores Cuadra Fernández
Luis Marino López Luque

Septiembre/2015

ÍNDICE

1	Introducción	3
1.1	Motivación del proyecto	3
1.2	Objetivos	4
1.3	Estructura del documento.....	4
2	Estado del Arte.....	6
2.1	Método Canvas	6
2.1.1	Relaciones con los Clientes	9
2.1.2	Canales de Comunicación	10
2.2	Programación orientada a eventos	10
2.2.1	Modelo Clásico vs. Eventos	11
2.2.2	La base del Interfaz de Usuario	11
2.2.3	Visual Basic	12
2.2.4	Estudio del Ambiente de Desarrollo.....	14
2.2.5	La opción elegida.....	27
2.3	Diseño Web Responsive	33
2.3.1	Generaciones del diseño Web.....	34
2.3.2	Estándares Web	40
2.3.3	HTML5: Nueva especificación de HTML	42
3	Propuestas	55
3.1	Consideraciones Generales	55
3.2	Software de gestión TMB	56
3.2.1	La Solicitud.....	56
3.2.2	Análisis del sistema de información.....	56

3.2.3	Diseño del Sistema de Información.....	74
3.2.4	Implementación del Sistema de Información	88
3.3	Libro de Mantenimiento online.....	97
3.3.1	La Solicitud.....	97
3.3.2	Análisis del Sistema de Información	97
3.3.3	Diseño del Sistema de Información.....	114
3.3.4	Implementación del Sistema de Información	129
4	Conclusiones y Líneas Futuras	141
4.1	Conclusiones	141
4.2	Líneas Futuras	142
5	Referencias.....	143
6	Anexos	144
6.1	Método Canvas TMB.....	144
6.2	Migración Access to MySql.....	145
6.3	Manual de Usuario Software de Gestión	152
6.4	Manual de Usuario Libro de Mantenimiento online	169
6.5	Backup Documento de Seguridad (Migración BBDD)	175
6.6	Glosario.....	180

1 Introducción

Todo proyecto tiene una motivación y unos objetivos a seguir para la consecución del mismo. Detallaremos en este capítulo tanto la motivación como los pasos a realizar para satisfacer dichos objetivos. Mostraremos a través del **Método Canvas**, la evolución del modelo de negocio de TECNOLOGÍA MECÁNICA BENITO SL, en adelante **TMB**, detallando el porqué de sus elecciones en tecnología, así como sus cambios en estrategia empresarial.

Continuaremos comentando la estructura del documento y mostrando un listado de acrónimos, que aparecerán a lo largo del texto, con sus correspondientes definiciones, y términos que se consideran necesarios para la correcta comprensión del contenido del documento.

1.1 Motivación del proyecto

La motivación principal de este PFC es garantizar la supervivencia y futuro lanzamiento empresarial de TMB, una PYME del sector de la automoción. En estos tiempos donde el término *competitividad* es conocido por todos, solo lograremos el objetivo mediante la utilización e inversión en tecnologías, tanto de información como de automoción en este caso, sin olvidar la continua formación del personal en dichas tecnologías.

En un entorno de crisis generalizado, donde la competitividad ha marcado el ritmo empresarial, la toma de decisiones en busca de ser la elección final del cliente es necesaria.

En este caso TMB, PYME situada al sur de la Comunidad de Madrid, convive con pequeñas empresas similares, concesionarios oficiales y grandes talleres *low cost*.

Ante este escenario, sin posibilidad de competir en precios finales, era necesario encontrar un servicio que ofrecer al cliente, por el que estuviera dispuesto a pagar y que eligiera a TMB. La empresa optó por el Método Canvas para el diseño de su modelo de negocio, donde la continua comunicación con el cliente a través de tecnologías de la información, pretende ser una de las claves para el éxito del modelo.

1.2 Objetivos

El propósito final es conseguir la fidelidad del cliente, gestionando de forma eficaz e individual la información que tenemos de él. Partiendo de un sistema de tratamiento de la información en papel, conseguimos a lo largo del proyecto informatizar la gestión de esta información y llegar a interactuar con el cliente a través de un portal Web, redes sociales, blog de encuentro... Por lo que nos encontramos ante un proyecto vivo, que evoluciona y se adecua a los cambios del mercado empresarial y marco social en el que trabaja la empresa.

Ante la importancia que tiene la información para conseguir la fidelidad del cliente, marcaremos **4 objetivos iterativos** a cumplir siempre que el escenario lo requiera.

- 1- Estudio del estado actual de los sistemas de tratamiento y obtención de la información de TMB.
- 2- Búsqueda de soluciones a las limitaciones que pueda generar dicha gestión y tratamiento de información.
- 3- Selección de las herramientas adecuadas para la gestión y tratamiento de la información.
- 4- Diseño y desarrollo de soluciones informáticas, utilizando las herramientas seleccionadas.

1.3 Estructura del documento

El documento se encuentra dividido en las siguientes secciones:

Capítulo 1. Introducción

La motivación y los objetivos a satisfacer son descritos en este capítulo. También esbozamos el escenario donde compite TMB para conocer sus necesidades y entender las decisiones adoptadas. Se finaliza mostrando un esquema de la memoria, describiendo brevemente el contenido de cada capítulo.

Capítulo 2. Estado del arte

A lo largo de este capítulo se establece el marco de trabajo. Conoceremos el Método Canvas y su importancia dentro de TMB. Nos adentraremos en la programación orientada a eventos,

necesaria para el desarrollo del software de gestión del Taller. Y por último, repasaremos la evolución del diseño Web y sus estándares, esbozando las características del Diseño Responsive, siempre poniendo el foco en HTML5.

Capítulo 3. Propuestas

El capítulo realiza una breve descripción inicial de las Fases de desarrollo y de los medios necesarios para la realización del proyecto. Continúa con el Análisis, Diseño e Implementación tanto del Programa de escritorio como del Portal Web.

Capítulo 4. Conclusiones y líneas futuras

Se exponen las conclusiones generales y personales obtenidas tras la realización del proyecto. Se finaliza este apartado con posibles trabajos futuros que se puedan realizar basándose en este proyecto.

Capítulo 5. Referencias

Se enumeran las referencias del proyecto y las fuentes bibliográficas.

Capítulo 6. Apéndices

Se anexan los siguientes apéndices a la memoria:

Método Canvas TMB (última versión)

Migración Access to MySQL

Manual de Usuario Software de Gestión

Manual de Usuario Zona Privada Portal Web

Capítulo Backup del Documento de Seguridad (Momento de Migración BD)

Glosario de Términos

2 Estado del Arte

A lo largo de este capítulo se establece el marco de trabajo. En primer lugar, describiremos el Modelo Canvas, Método escogido por TMB para el diseño de su Modelo de Negocio. En segundo lugar, nos adentraremos en la programación orientada a eventos, paradigma de la programación necesaria para el desarrollo del software de gestión del Taller. Y por último, repasaremos la evolución del diseño Web y sus estándares, poniendo el foco en HTML5, quinta revisión importante de HTML, lenguaje de marcas para la elaboración de páginas Web.

2.1 Método Canvas

La mejor manera de describir un modelo de negocio es dividirlo en nueve módulos básicos que reflejen la lógica que sigue una empresa para conseguir ingresos. Estos nueve módulos cubren las cuatro áreas principales de un negocio: clientes, oferta, infraestructuras y viabilidad económica. [1]

El Modelo Canvas permite diseñar de forma rápida un modelo de negocio permitiendo plasmar en una sola hoja toda tu idea. Integrado por nueve apartados, el Método *Lienzo* se puede resumir en cuatro pasos:

1. conocer al cliente.
2. identificar qué problemas o necesidades tiene
3. definir una solución
4. saber cuánto estaría dispuesto a pagar por ello.

Business Model Canvas

Diseñado para:

Diseñado por:

Et:

Iteración:

<p>Socios clave</p> <p>¿quién son nuestros socios clave? ¿quién son nuestros proveedores de recursos? ¿quién son nuestros socios de distribución? ¿quién son nuestros socios de canales? ¿quién son nuestros socios de recursos? ¿quién son nuestros socios de canales? ¿quién son nuestros socios de recursos?</p> <p>Beneficios para socios: - Reducción de costos - Acceso a nuevos mercados - Acceso a nuevos recursos</p>	<p>Actividades clave</p> <p>¿qué actividades clave requiere nuestra propuesta de valor? ¿qué actividades clave requiere nuestra propuesta de valor? ¿qué actividades clave requiere nuestra propuesta de valor? ¿qué actividades clave requiere nuestra propuesta de valor? ¿qué actividades clave requiere nuestra propuesta de valor?</p> <p>Categorías: - Operación - Logística - Personal - Tecnología</p>	<p>Propuestas de valor</p> <p>¿qué valor entregamos al cliente? ¿qué valor entregamos al cliente? ¿qué valor entregamos al cliente? ¿qué valor entregamos al cliente? ¿qué valor entregamos al cliente?</p> <p>Beneficios: - Reducción de costos - Acceso a nuevos mercados - Acceso a nuevos recursos - Acceso a nuevos canales - Acceso a nuevos socios</p>	<p>Relaciones con clientes</p> <p>¿qué tipo de relación queremos que establezcamos y mantengamos cada uno de nuestros segmentos de clientes? ¿qué tipo de relación queremos que establezcamos y mantengamos cada uno de nuestros segmentos de clientes? ¿qué tipo de relación queremos que establezcamos y mantengamos cada uno de nuestros segmentos de clientes?</p> <p>Tipos de relaciones: - Personal - Personalizado - Personalizado - Personalizado - Personalizado</p>	<p>Segmentos de cliente</p> <p>¿para qué segmentos creamos valor? ¿para qué segmentos creamos valor? ¿para qué segmentos creamos valor? ¿para qué segmentos creamos valor? ¿para qué segmentos creamos valor?</p> <p>Beneficios: - Reducción de costos - Acceso a nuevos mercados - Acceso a nuevos recursos - Acceso a nuevos canales - Acceso a nuevos socios</p>
<p>Recursos clave</p> <p>¿qué recursos clave requiere nuestra propuesta de valor? ¿qué recursos clave requiere nuestra propuesta de valor? ¿qué recursos clave requiere nuestra propuesta de valor? ¿qué recursos clave requiere nuestra propuesta de valor? ¿qué recursos clave requiere nuestra propuesta de valor?</p> <p>Tipos de recursos: - Operación - Logística - Personal - Tecnología</p>		<p>Canales</p> <p>¿a través de qué canales queremos que nuestros segmentos de clientes accedan a nuestra propuesta de valor? ¿a través de qué canales queremos que nuestros segmentos de clientes accedan a nuestra propuesta de valor? ¿a través de qué canales queremos que nuestros segmentos de clientes accedan a nuestra propuesta de valor?</p> <p>Tipos de canales: - Personal - Personalizado - Personalizado - Personalizado - Personalizado</p>		
<p>Estructura de costes</p> <p>¿cuáles son los costos más importantes relacionados a nuestra propuesta de valor? ¿cuáles son los costos más importantes relacionados a nuestra propuesta de valor? ¿cuáles son los costos más importantes relacionados a nuestra propuesta de valor? ¿cuáles son los costos más importantes relacionados a nuestra propuesta de valor? ¿cuáles son los costos más importantes relacionados a nuestra propuesta de valor?</p> <p>Tipos de estructura de costes: - Operación - Logística - Personal - Tecnología</p>		<p>Fuentes de ingresos</p> <p>¿cómo queremos cobrar a nuestros segmentos de clientes? ¿cómo queremos cobrar a nuestros segmentos de clientes? ¿cómo queremos cobrar a nuestros segmentos de clientes? ¿cómo queremos cobrar a nuestros segmentos de clientes? ¿cómo queremos cobrar a nuestros segmentos de clientes?</p> <p>Tipos de fuentes de ingresos: - Operación - Logística - Personal - Tecnología</p>		

Ilustración 1 Modelo Canvas

El Modelo Canvas transforma el enciclopédico Plan de Negocio en un recuadro dividido en nueve celdas que los emprendedores deben rellenar con post-its. El objetivo de Osterwalder es que la información se pueda cambiar rápida y fácilmente para adaptarse a los continuos cambios del mercado. [2]

Ahora las empresas están en continua transformación y la clave para crecer ya no consiste en tener más clientes, sino en ofrecer lo que éste busca. Esto obliga al empresario a estar en contacto continuo con el cliente para conocer su opinión.

Los pasos del Método Canvas: [2]

1. **Segmentos de mercado:** en este módulo se definen los diferentes grupos de personas o entidades a los que se dirige una empresa.
2. **Propuestas de valor:** se describe el conjunto de productos y servicios que crean valor para un segmento de mercado específico. La propuesta de valor es el factor que hace que un cliente se decante por una u otra empresa; su finalidad es solucionar un problema o satisfacer una necesidad del cliente.
3. **Canales:** explica el modo en que una empresa se comunica con los diferentes segmentos de mercado para llegar a ellos y proporcionarles una propuesta de valor. Los canales de comunicación, distribución y venta establecen el contacto entre la empresa y los clientes.
4. **Relaciones con clientes:** Las empresas deben definir el tipo de relación que desean establecer con cada segmento de mercado. La relación puede ser personal o automatizada.
5. **Fuentes de ingresos:** Si los clientes constituyen el centro de un modelo de negocio, las fuentes de ingresos son sus arterias. Las empresas deben preguntarse lo siguiente: ¿por qué valor está dispuesto a pagar cada segmento de mercado?
6. **Recursos clave:** se describen los activos más importantes para que un modelo de negocio funcione. Cada modelo de negocio requiere recursos clave diferentes. Un fabricante de microchips necesita instalaciones de producción con un capital elevado, mientras que un diseñador de microchips depende más de los recursos humanos.
7. **Actividades clave:** se describen las acciones más importantes que debe emprender una empresa para que su modelo de negocio funcione.
8. **Asociaciones clave:** se describe la red de proveedores y socios que contribuyen al funcionamiento de un modelo de negocio.
9. **Estructura de costes:** En este último módulo se describen todos los costes que implica la puesta en marcha de un modelo de negocio.

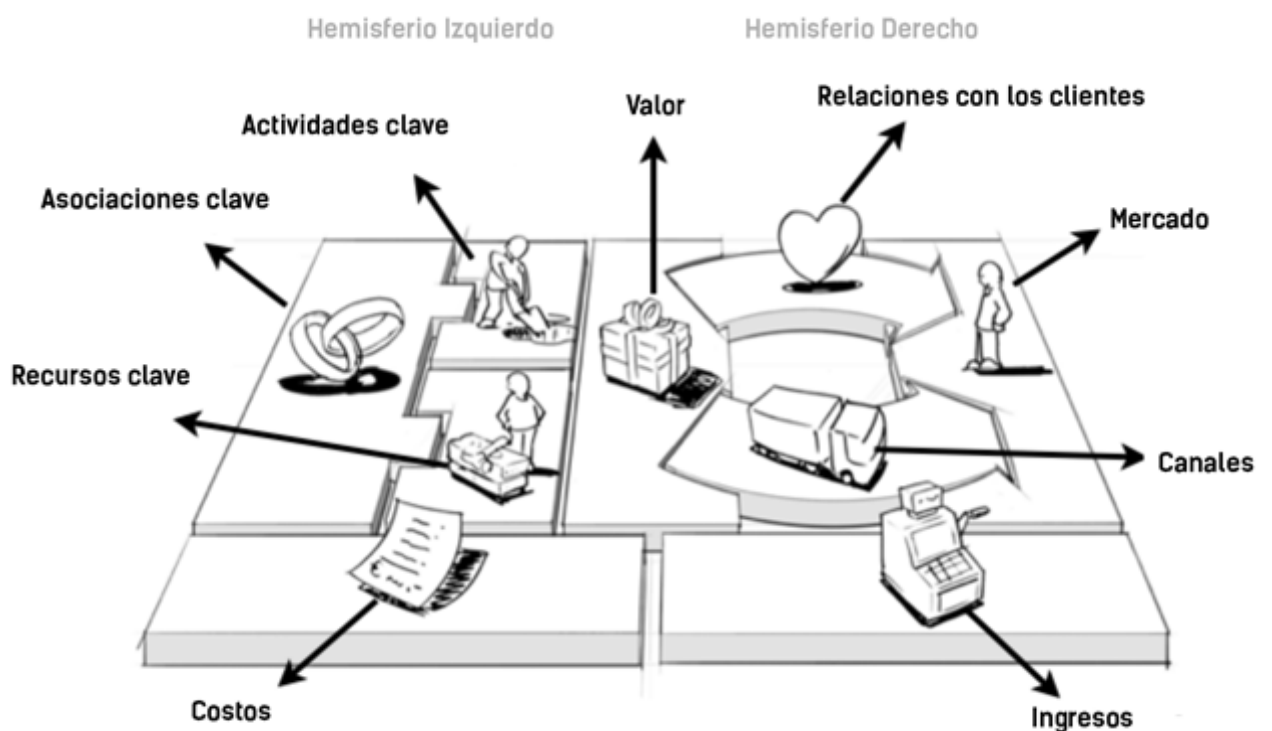


Ilustración 2 Modelo de Negocio

Dejando a un lado los aspectos técnicos y las actividades clave en el ámbito de la automoción, centraremos nuestra atención en dos aspectos principalmente:

1. Las relaciones con los clientes
2. Los canales para conseguir esa comunicación

2.1.1 Relaciones con los Clientes

¿Para quienes estamos creando valor? ¿Quiénes son nuestros clientes más importantes? ¿Qué tipo de relación tenemos con nuestros clientes? ¿Y cómo mantenerla a lo largo del tiempo?

Debemos tener claro como conseguiremos atraer a nuevos clientes, mantenerlos y hacerlos crecer. La táctica principal de TMB consistirá en una **atención personalizada** del cliente, ofreciendo a su vez **un trato rápido y eficaz**. TMB garantizará un **servicio perfecto pre/post reparación**. El cliente que hable con nosotros debe saber que habla con un experto y que le vamos a asesorar de una manera honesta sobre la reparación de su coche.

A través de esta línea de trabajo conseguiremos la fidelidad del cliente y que vuelvan al taller en futuras reparaciones. A su vez *el boca a boca* de los clientes satisfechos atraerá con toda seguridad a nuevos clientes.

2.1.2 Canales de Comunicación

¿Cómo conecto con nuestros clientes? ¿Cuáles funcionan mejor? ¿Cuáles son los más rentables? ¿Cómo lo integramos en las rutinas de nuestros clientes?

La relación que establezca con el cliente debe ser duradera y debemos definir como queremos que nuestro producto llegue al cliente.

El servicio personalizado puede llegar a través del **Portal Web** y su libro de mantenimiento online. **El blog de encuentro** Diagnostico Avería mostrará al cliente el día a día del trabajo en su Taller. No podemos olvidar tampoco el poder de las **RRSS** con sus avisos generales que atraerán también la atención del cliente.

En último lugar contactaremos de forma directa con el cliente mediante **Mail ó teléfono**, teniendo en cuenta siempre la adecuación del servicio a los dispositivos móviles.

2.2 Programación orientada a eventos

La programación dirigida por eventos es un paradigma de programación (Imperativa) en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen. [4]

Al contrario de la programación estructurada, donde el programador define el flujo del programa, en la programación orientada a eventos el usuario tendrá un papel primordial, ya que será él quien dirija el flujo de la aplicación. Queda entonces clara la necesidad de una interfaz gráfica para interactuar con el usuario.

La interacción con un programa a través de una interfaz gráfica, simplifica la labor del usuario, pero complica la del programador. En su implementación podemos distinguir dos tareas fundamentales: diseño y programación. El diseño supone determinar la cantidad de componentes, su agrupación y su ubicación en el espacio, mientras que la programación supone

determinar las acciones que se llevarán a cabo cuando el usuario interactúe con cada componente. [3]

2.2.1 Modelo Clásico vs. Eventos

En los primeros tiempos de la computación, los programas eran secuenciales, también llamados Batch. Un programa secuencial arranca, lee parámetros de entrada, procesa estos parámetros, y produce un resultado, todo de manera lineal y sin intervención del usuario mientras se ejecuta.

En contraposición al modelo clásico, la programación orientada a eventos permite interactuar con el usuario en cualquier momento de la ejecución. Esto se consigue debido a que los programas creados bajo esta arquitectura se componen por un bucle exterior permanente encargado de recoger los eventos, y distintos procesos que se encargan de tratarlos. Habitualmente, este bucle externo permanece oculto al programador que simplemente se encarga de tratar los eventos, aunque en algunos entornos de desarrollo (IDE) será necesaria su construcción.

La programación orientada a eventos supone una complicación añadida con respecto a otros paradigmas de programación, debido a que el flujo de ejecución del software escapa al control del programador. En cierta manera podríamos decir que en la programación clásica el flujo estaba en poder del programador y era este quien decidía el orden de ejecución de los procesos, mientras que en programación orientada a eventos, es el usuario el que controla el flujo y decide. [4]

2.2.2 La base del Interfaz de Usuario

La programación dirigida por eventos es la base de lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo. Al comenzar la ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento. Cuando alguno de los eventos esperados por el programa tenga lugar, el programa pasará a ejecutar el código correspondiente.

Con la evolución de los lenguajes orientados a eventos, la interacción del software con el usuario ha mejorado enormemente permitiendo la aparición de interfaces que, aparte de ser la vía de

comunicación del programa con el usuario, son la propia apariencia del mismo. Estas interfaces, también llamadas GUI (Graphical User Interface), han sido la herramienta imprescindible para acercar la informática a los usuarios. Los periféricos cobran una gran importancia tras la aparición de este tipo de programas, sin olvidar la aparición de los sistemas operativos modernos con sus interfaces gráficas.

Tecnología Mecánica Benito -- PANTALLA PRINCIPAL --

Archivo Gestor BBDD Ayuda

Cliente

DNI 52981185H Nombre ALFREDO Apellido 1 FERNANDEZ Apellido 2 ARROYO
Dirección LOS PINOS 11 Localidad VILLALUENGA DE L Provincia TOLEDO CP 45520
Email ARMANDO -- 627050372 Teléfono 1 627825640 Teléfono 2 647410099

Vehículos

Matrícula 8610GPB Marca PEUGEOT Modelo 308 SPORT HDI Tipo Motor D/9HZ Año 05/08/2009 Ncc 4/1560 PF 11,45 Kw 80 Chasis VF34C9HZC9S131846

Vista Rápida

Fecha Reservada 21/03/2016 Prox. Revisión 107500
Prox. ITV 20/09/2015 Prox. Distribución 180000

F. Combustible NO HAY DATOS F. Aire C3282 F. Aceite HU716/2X F. Habitaculo CU2940 Aceite Motor 10W40 Aceite Cambio 75W90 Bujías NO HAY DATOS

Neumáticos 205/55 R16 91V

Código Radio NO DATOS Código Llave NO DATOS

Ref. Distribución NO HAY DATOS 180000 KM. O,10 AÑOS (4/2019)

21:16:30 BÚSQUEDA POR DNI: 52981185H 27/07/2015

Ilustración 3 Interfaz TMB

Con el paso del tiempo, aparecieron herramientas que incluían código que automatiza parte de las tareas más comunes en la detección y tratamiento de eventos. Hay que destacar los entornos de programación visual que conjugaban una herramienta de diseño gráfica para la GUI y un lenguaje de alto nivel. Entre estas herramientas se encontraba Visual Basic.

2.2.3 Visual Basic

Este lenguaje de programación dirigido por eventos tuvo como última versión la 6.0, utilizada para el desarrollo de nuestro programa de gestión y con soporte hasta 2008. La versión 6.0, que

puede generar código ejecutable directo en 32 bits, continúa aun utilizándose masivamente, y es compatible con las últimas versiones de los sistemas Windows, como Windows 7 y Windows 8.

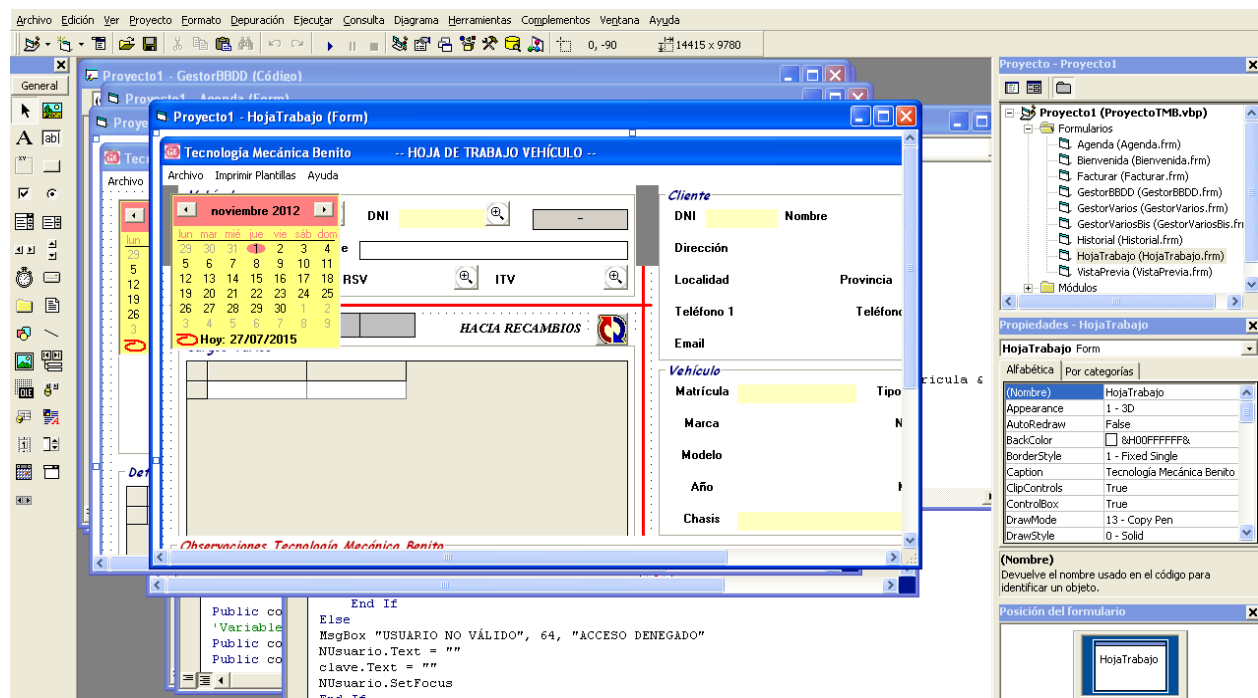


Ilustración 4 IDE Visual Basic

Visual Basic contiene un IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI. Aunque como vemos mantiene su propio entorno de desarrollo, estamos obligados a hacer una breve exposición de Visual Studio.

2.2.3.1 Visual Studio

Microsoft Visual Studio es un entorno para sistemas operativos Windows cuya finalidad es la creación de aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte .NET.

El objetivo de esta colección de herramientas es proporcionar al equipo de programación una sola fuente y un solo marco de trabajo para todo el desarrollo. Algunas de las funciones se solapan, pero las herramientas están diseñadas para complementarse entre sí y proporcionar un entorno completo para diseño, desarrollo y despliegue. [6]

Las herramientas de la *suite* están concebidas para satisfacer las diferentes necesidades de desarrollo de distintos tipos de programadores, incluyendo los desarrolladores de Web, de aplicaciones, de empresa y de contenido.

2.2.4 Estudio del Ambiente de Desarrollo

En el siguiente apartado vamos a identificar, mediante un pequeño estudio, conceptos de programación orientada al evento confrontándola con lenguajes visuales y orientados a objetos, asimismo, *pondremos la primera piedra* en el manejo del ambiente de desarrollo de Visual Basic en la construcción de interfaces gráficas. [5]

2.2.4.1 Bases del estudio

Los lenguajes visuales orientados al evento y con manejo de componentes, dan al usuario que no cuenta con mucha experiencia en desarrollo, la posibilidad de construir sus propias aplicaciones utilizando interfaces gráficas sobre la base de ocurrencia de eventos.

Para soportar este tipo de desarrollo interactúan dos tipos de herramientas, una que permite realizar diseños gráficos y, un lenguaje de alto nivel que permite codificar los eventos. Con dichas herramientas es posible desarrollar cualquier tipo de aplicaciones basadas en el entorno.

Los programadores expertos y novatos aprecian el lenguaje (Visual Basic), los primeros por la facilidad con la que desarrollan aplicaciones complejas en poquísimo tiempo (comparado con lo que cuesta programar en Visual C++, por ejemplo), y en el caso de los programadores novatos, por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje. El precio que hay que pagar por utilizar Visual Basic es una menor velocidad o eficiencia en las aplicaciones.

Visual Basic es un lenguaje de programación visual, también llamado lenguaje de 4ta. Generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla. También es un programa basado en objetos, aunque no orientado a objetos como Visual C++. La diferencia está en Visual Basic utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y poliformismo propios de los lenguajes orientado a objetos como Java y C++.

2.2.4.2 Programas secuenciales, interactivos y orientados a eventos

En los primeros tiempos de los ordenadores los programas eran de tipo secuencial (también llamados tipo batch). Un **programa secuencial** es un programa que se arranca, lee los datos que necesita, realiza los cálculos e imprime o guarda en el disco los resultados. De ordinario, mientras un programa secuencial está ejecutándose no necesita ninguna intervención del usuario. A este tipo de programas se les llama también programas basados u orientados a procedimientos o a algoritmos.

Los **programas interactivos** exigen la intervención del usuario en tiempo de ejecución, bien para suministrar datos, bien para indicar al programa lo que debe hacer por medio de menús. Los programas interactivos limitan y orientan la acción del usuario.

Por su parte los **programas orientados a eventos** son los programas típicos de Windows, tales como Word, Excel, PowerPoint... Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de las acciones del usuario, que en este caso son llamadas eventos.

Estos programas pasan la mayor parte de su tiempo esperando las acciones del usuario (**eventos**) y respondiendo a ellas. Las acciones que el usuario puede realizar en un momento determinado son variadísimas, y exigen un tipo especial de programación: **la programación orientada a eventos**. Este tipo de programación es sensiblemente más complicada que la secuencial y la interactiva, pero con los lenguajes visuales de hoy, se hace sencilla y agradable.

2.2.4.3 Eventos

Ya se ha dicho que las acciones del usuario sobre el programa se llaman eventos. Son eventos típicos el *click* sobre un botón, el hacer *doble click* sobre el nombre de un fichero para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón.

Cada vez que se produce un evento sobre un determinado tipo de control, Visual Basic arranca una determinada función o procedimiento que realiza la acción programada por el usuario para ese evento concreto. Estos procedimientos se llaman con un nombre que se forma a partir del nombre del objeto y el nombre del evento, separados por el carácter (`_`), como por

ejemplo *txtBox_click*, que es el nombre del procedimiento que se ocupará de responder al evento *click* en el objeto *txtBox*.

2.2.4.4 Propiedades y Métodos

Además de los eventos, la mayor parte de los objetos, como los formularios y los controles, son suministrados con propiedades y métodos.

Una **propiedad** es una asignación que describe algo sobre un objeto como un formulario. Dependiendo de la propiedad, se la puede asignar en tiempo de diseño usando la ventana Propiedades y/o en tiempo de ejecución al programar. Por ejemplo, *MinButton* es una propiedad que puede asignarse como TRUE (verdadero) o FALSE (falso). Dependiendo de la asignación, el formulario tendrá o no tendrá un botón minimizar.

Los **métodos** son funciones que también son llamadas desde programa, pero a diferencia de los procedimientos no son programadas por el usuario, sino que vienen ya *pre-programadas* con el lenguaje. Los métodos realizan tareas típicas, previsibles y comunes para todas las aplicaciones, de ahí que vengan con el lenguaje y que se libere al usuario de la tarea de programarlos. Cada tipo de objeto o de control tiene sus propios métodos.

En general solo pueden ser ejecutados en tiempos de ejecución no en tiempo de diseño. Algunos ejemplos de métodos de formularios son el método MOVE, que mueve un formulario en un espacio de dos dimensiones en la pantalla, y el método ZORDER que sitúa el formulario delante o detrás de otras ventanas.

Los métodos son invocados dando nombres al objeto y cuyo método se está llamando, listando el operador punto (.), y después listando el nombre del método. Por ejemplo, *Form1.Show 1*, este método *Show* carga y muestra un formulario, dos acciones distintas que forman ambas partes del proceso de nacimiento o inicio a un formulario (al ejecutar el formulario de inicio se carga también primero el formulario y después se muestra).

2.2.4.5 Visual significa Visual

Como el nombre lo indica, una gran parte de la programación con Visual Basic se realiza visualmente. Esto significa que durante el tiempo de diseño podemos ver la forma en el que el programa se verá al ejecutarse. Esta es una gran ventaja sobre otros lenguajes de programación

debido a que se tiene la capacidad de cambiar y experimentar con el diseño hasta que se esté satisfecho con los colores, proporciones e imágenes que incluya en su programa.

Visual Basic está orientado a la realización de programas para Windows, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc.

2.2.4.6 Modo de diseño y Modo de ejecución

La aplicación Visual Basic de *Microsoft* puede trabajar de dos modos distintos: en modo de diseño y en modo de ejecución.

En **modo de diseño** el usuario construye interactivamente la aplicación, colocando controles en el formulario, definiendo sus propiedades, desarrollando funciones para gestionar los eventos, y la aplicación se prueba en **modo de ejecución**.

En ese caso el usuario actúa sobre el programa (introduce eventos) y prueba cómo responde el programa. Hay algunas propiedades de los controles que deben establecerse en modo de diseño, pero muchas otras pueden cambiarse en tiempo de ejecución desde el programa escrito en Visual Basic. También hay propiedades que sólo pueden establecerse en modo de ejecución y que no son visibles en modo de diseño.

2.2.4.7 Formularios y Controles

Elementos gráficos como los botones, las cajas de diálogo y de texto, las cajas de selección desplegables, los botones de opción y de selección, las barras de desplazamiento horizontal y vertical, los gráficos, los menús, y muchos otros tipos de elementos son **controles** para Visual Basic. Cada control debe tener un nombre a través del cual se puede hacer referencia a él en el programa. Visual Basic proporciona nombres por defecto que el usuario puede modificar.

En la terminología de Visual Basic se llama **formulario** (*form*) a una ventana. Un formulario puede ser considerado como una especie de contenedor para los controles. Una aplicación puede tener varios formularios, pero un único formulario puede ser suficiente para las aplicaciones más sencillas.

2.2.4.8 Objetos y Propiedades

Los formularios y los distintos tipos de controles son entidades genéricas de las que puede haber varios ejemplares concretos en cada programa. En programación orientada a objetos (más bien basada en objetos, habría que decir) se llama clase a estas entidades genéricas, mientras que se llama **objeto** a cada ejemplar de una clase determinada. Por ejemplo, en un programa pueden haber varios botones, cada uno de los cuales es uno objeto del tipo de control *command button*, que sería la clase.

Cada formulario y cada tipo de control tienen un conjunto de propiedades que definen su aspecto gráfico (tamaño, color, posición en la ventana, tipo y tamaño de letra, etc.) y su forma de responder a las acciones del usuario (si está activo o no, por ejemplo). Cada propiedad tiene un nombre que viene ya definido por el lenguaje.

Por lo general, las **propiedades** de un objeto son datos que tienen valores lógicos (true, false) o numéricos concretos, propios de ese objeto y distintos de las de otros objetos de su clase. Así pues, cada clase, tipo de objeto o control tiene su conjunto de propiedades, y cada objeto o control concreto tiene unos valores determinados para las propiedades de su clase.

Casi todas las propiedades de los objetos pueden establecerse en tiempo de diseño y también - casi siempre- en tiempo de ejecución. En este segundo caso se accede a sus valores por medio de las sentencias del programa, en forma análoga a como se accede a cualquier variable en un lenguaje de programación. Para ciertas propiedades ésta es la única forma de acceder a ellas.

Se puede acceder a una propiedad de un objeto por medio del nombre del objeto a que pertenece, seguido de un punto y el nombre de la propiedad, como por ejemplo *optColor.objName*.

2.2.4.9 Orden de Disparo de Eventos

Para controlar con éxito la aparición y el comportamiento de los formularios (y también de los controles) en tiempos de ejecución, debe comprenderse en qué **orden se disparan los eventos**. Las consideraciones del orden de disparo de los eventos deciden generalmente por el usuario donde debe ser colocada una parte determinada de código de respuesta de un evento. Los eventos de formularios pueden ser divididos en los grupos siguientes:

1.- Inicio.

2.- Respuesta a una acción (de usuario).

3.- Vinculación.

4.- Cierre.

Es importante también comprender que un evento inicia automáticamente con frecuencia a otro evento, produciendo un efecto en cascada. Por ejemplo un evento *KeyPress* no puede ser disparada sin disparar también los eventos *KeyUp* y *KeyDown*. El secreto para trabajar con esta clase de situaciones es una comprensión clara de que es lo que dispara cada evento en la secuencia; el peligro de la codificación es iniciar una cadena sin fin de llamada a eventos circulares recursivos.

2.2.4.10 Eventos Generales

Carga y Descarga de Formularios

Cuando se arranca una aplicación, o más en concreto cuando se visualiza por primera vez un formulario se producen varios eventos consecutivos: ***Initialize, Load, Activate y Paint***. Cada uno de estos eventos se puede aprovechar para realizar ciertas operaciones por medio de la función correspondiente.

Para inicializar las variables definidas a nivel de módulo se suele utilizar el evento *Initialize*, que tiene lugar antes que el ***Load***. El evento *Load* se activa al cargar un formulario. Con el formulario principal esto sucede al arrancar la ejecución de un programa; con el resto de los formularios al mandarlos cargar desde cualquier procedimiento o al hacer referencia a alguna propiedad o control de un formulario que no esté cargado. Al descargar un formulario se produce el evento ***Unload***. Si se detiene el programa desde el botón Stop de Visual o con un *End*, no se pasa por el evento *Unload*. Para pasar por el evento *Unload* es necesario cerrar la ventana con el botón de cerrar o llamarlo explícitamente. El evento *QueryUnload* se produce antes del evento *Unload* y permite por ejemplo enviar un mensaje de confirmación.

QueryUnload, se dispara por un evento *Unload* de formulario, antes de que se ejecute ningún código del evento *Unload*. *QueryUnload* da la oportunidad de detener la descarga del formulario de la memoria cuando sea apropiado, por ejemplo, cuando el usuario haya cambiado valores del formulario sin guardar los cambios. El formulario no será descargado si la variable *Cancel* está

asignada como *True* en el procedimiento del evento *QueryUnload*. Es una práctica común asignar esta variable como *True* a continuación de una consulta del usuario a lo largo de las líneas de "¿Guardar los cambios? Sí, No, Cancelar" cuando el usuario responde cancelando. La función *MsgBox* puede usarse fácilmente para dar al usuario una oportunidad de cancelar una descarga.

Unload, se dispara cuando un usuario cierra el formulario usando el comando Cerrar en el menú Control, ó cuando se ejecuta en código un método *Unload*. *Unload* dispara inmediatamente un evento *QueryUnload* como acabamos de describir. Se puede usar el evento *Unload* (así como *QueryUnload*) para realizar tareas de cierre como guardar y convalidar datos.

Terminate, se produce cuando se eliminan de la memoria todas las referencias a una instancia de un formulario.

El evento ***Load*** de un formulario se suele utilizar para ejecutar una función que dé valor a sus propiedades y a las de los controles que dependen de dicho formulario. No se puede utilizar para dibujar o imprimir sobre el formulario, pues en el momento en que se produce este evento el formulario todavía no está disponible para dichas operaciones.

Se puede ocultar un formulario sin descargarlo con el método *Hide* o haciendo la propiedad *Visible = False*. Esto hace que el formulario desaparezca de la ventana, aunque sus variables y propiedades sigan estando accesibles y conservando sus valores. Para hacer visible un formulario oculto pero ya cargado se utiliza el método *Show*, que equivale a hacer la propiedad *Visible = True*, y que genera los eventos *Activate* y *Paint*. Si el formulario no había sido cargado previamente, el método *Show* genera los cuatro eventos mencionados.

Cuando un formulario pasa a ser la ventana activa se produce el evento *Activate* y al dejar de serlo el evento *Deactivate*. En el caso de que el formulario que va a ser activo no estuviera cargado ya, primero sucederían los eventos *Initialize*, *Load* y luego los eventos *Activate* y *Paint*.

El Foco (focus)

En todas las aplicaciones de Windows, en cualquiera de sus versiones, siempre hay un único control, formulario o ventana que puede recibir clicks del ratón o entradas desde teclado. En cada momento ese control, ventana o formulario es el que dispone del "foco" (*focus*).

Este foco puede ser trasladado de un objeto a otro por código o por interacciones del usuario, como por ejemplo seleccionando con el ratón en distintos puntos de la pantalla o pulsando la tecla Tab. Cada vez que un objeto pierde el foco se produce su evento *LostFocus* y, posteriormente, el evento *GotFocus* del objeto que ha recibido el foco.

Dos propiedades de muchos controles relacionadas con el foco son *TabIndex* y *TabStop*. ***TabStop*** determina si el foco se va o no a posar en el objeto al pulsar la tecla Tab (si *TabStop* está a False no se puede obtener el foco mediante el tabulador) y *TabIndex* determina el orden en el que esto va a suceder. Así al cargar un formulario, el foco estará en aquel objeto cuyo *TabIndex* sea **0**.

Al pulsar la tecla Tab el foco irá al objeto que tenga ***TabIndex* = 1** y así sucesivamente. Para retroceder en esta lista se pulsa Mayúsculas+Tab. La propiedad *TabIndex* se puede determinar en tiempo de diseño por medio de la caja de propiedades, del modo habitual. Cuando a un control se le asigna un determinado valor de *TabIndex*, Visual Basic ajusta automáticamente los valores de los demás controles. Para que un formulario reciba el foco es necesario que no haya en él ningún control que sea capaz de recibirlo. Un grupo de botones de opción tiene un único *TabIndex*, es decir, se comporta como un único control.

KeyPress, KeyUp y KeyDown

El evento ***KeyPress*** sucede cuando el usuario pulsa y suelta determinada tecla. En este evento el único argumento *KeyAscii* es necesario para conocer cuál es el código ASCII de la tecla pulsada. El evento ***KeyDown*** se produce cuando el usuario pulsa determinada tecla y el evento ***KeyUp*** al soltar una tecla.

Los eventos *KeyUp* y *KeyDown* tienen un segundo argumento llamado *Shift* que permiten determinar si esa tecla se ha pulsado estando pulsadas a la vez cualquier combinación de las teclas.

2.2.4.11 Eventos Relacionados con el Ratón

Click y DbClick

El evento ***Click*** se activa cuando el usuario pulsa y suelta rápidamente uno de los botones del ratón. También puede activarse desde código (sin tocar el ratón) variando la propiedad *Value* de

uno de los controles. En el caso de un formulario este evento se activa cuando el usuario clicca sobre una zona del formulario en la que no haya ningún control o sobre un control que en ese momento esté inhabilitado (propiedad *Enabled = False*). En el caso de un control, el evento se activa cuando el usuario realiza una de las siguientes operaciones:

- ☐ *Clicar* sobre un control con el botón derecho o izquierdo del ratón. En el caso de un botón de comando, de un botón de selección o de un botón de opción, el evento sucede solamente al darle un *click* con el botón izquierdo.

- ☐ Seleccionar un registro de alguno de los varios tipos listas desplegables que dispone Visual Basic.

- ☐ Pulsar la barra espaciadora cuando el foco está en un botón de comando, en un botón de selección o en un botón de opción.

- ☐ Pulsar la tecla Return cuando en un formulario hay un botón que tiene su propiedad *Default = True*.

- ☐ Pulsar la tecla Esc cuando en un formulario hay un botón que tiene su propiedad *Cancel = True*.

- ☐ Pulsar una combinación de teclas aceleradoras (Alt + otra tecla, como por ejemplo cuando se despliega el menú File de Word con Alt+F) definidas para activar un determinado control de un formulario.

También se puede activar el evento *Click* desde código realizando una de las siguientes operaciones:

- ☐ Hacer que la propiedad *Value* de un botón de comando valga *True*.

- ☐ Hacer que la propiedad *Value* de un botón de opción valga *True*.

- ☐ Modificar la propiedad *Value* de un botón de selección.

El evento ***DbClick*** sucede al *clicar* dos veces seguidas sobre un control o formulario con el botón izquierdo del ratón.

MouseDown, MouseUp y MouseMove

El evento **MouseDown** sucede cuando el usuario pulsa cualquiera de los botones del ratón, mientras que el evento **MouseUp** sucede al soltar un botón que había sido pulsado. El evento **MouseMove** sucede al mover el ratón sobre un control o formulario.

Los eventos **MouseUp** y **MouseDown** tienen algunos argumentos que merecen ser comentados. El argumento **Button** indica cuál de los botones del ratón ha sido pulsado o soltado, y el argumento **Shift** indica si además alguna de las teclas alt, shift o ctrl está también pulsada.

DragOver y DragDrop

El evento **DragOver** sucede mientras se está arrastrando un objeto sobre un control. Suele utilizarse para variar la forma del cursor que se mueve con el ratón dependiendo de si el objeto sobre el que se encuentra el cursor en ese momento es válido para soltar o no. El evento **DragDrop** sucede al concluir una operación de arrastrar y soltar.

2.2.4.12 Eventos Y Propiedades relacionadas con Gráficos

El Evento Paint

El evento **Paint** se ejecuta cuando un objeto -de tipo *form* o *pictureBox*- se hace visible. Su finalidad es que el resultado de los métodos gráficos y del método *print* aparezca en el objeto correspondiente. Hay que tener en cuenta que si se introducen métodos gráficos en el procedimiento *form_load* su resultado no aparece al hacerse visible el formulario (es como si se dibujara sobre el formulario antes de que éste existiera). Para que el resultados de *print* y de los métodos gráficos aparezcan al hacerse visible el formulario, deben introducirse en el procedimiento *paint_form*.

También los controles *pictureBox* tienen evento *paint*, que se ejecuta al hacerse visibles. El evento **Paint** tiene mucha importancia en relación con el refresco de los gráficos y con la velocidad de ejecución de los mismos.

La Propiedad Drawmode

Esta es una propiedad bastante importante y difícil de manejar, sobre todo si se quieren realizar cierto tipo de acciones con los métodos gráficos. La propiedad **DrawMode** controla cómo se dibujan los controles *line* y *shape*, así como los resultados de los métodos gráficos *pset*, *line* y *circle*. La opción por defecto hace que cada elemento gráfico se dibuje con el color correspondiente (por defecto el *foreColor*) sobre lo dibujado anteriormente. En ocasiones esto no es lo más adecuado pues, por ejemplo, si se superponen dos figuras del mismo color o si se dibuja con el *backColor*, los gráficos resultan indistinguibles.

La Propiedad Autoredraw

Esta propiedad tiene una gran importancia. En principio, todas las aplicaciones de Windows permiten superponer ventanas y/u otros elementos gráficos, recuperando completamente el contenido de cualquier ventana cuando ésta se selecciona de nuevo y viene a primer plano (es la ventana activa). A esto se llama redibujar (redraw) la ventana. Cualquier aplicación que se desarrolle en Visual Basic debe ser capaz de *redibujarse* correctamente, pero para ello el programador debe conocer algo de la propiedad **AutoRedraw**.

Por defecto, Visual Basic *redibuja* siempre los controles que aparecen en un formulario. Esto no sucede sin embargo con el resultado de los métodos gráficos y de *print*. Para que la salida de estos métodos se redibuje es necesario adoptar uno de los dos métodos siguientes:

1.- Si en el *form* o *pictureBox* la propiedad **AutoRedraw** está en false:

☐ Si los métodos gráficos y *print* están en el procedimiento correspondiente al evento *paint* se redibujan en el plano de fondo (los métodos vuelven a ejecutarse, por lo que el proceso puede ser lento en ciertos casos).

☐ Si los métodos gráficos y *print* están fuera del evento *paint* no se *redibujan*.

2.- Si en el *form* o *pictureBox* la propiedad **AutoRedraw** está en true

☐ Si los métodos gráficos y *print* están en el evento *paint* se ignoran.

□ Si los métodos gráficos y *print* están fuera del evento *paint* se *redibujan* guardando en memoria una copia de la zona de pantalla a refrescar. Este es la forma más rápida de conseguir que los gráficos y el texto se *redibujen*. Tiene el inconveniente de necesitar más memoria.

La propiedad *AutoRedraw* de los *forms* y de las *pictureBox* es independiente, por lo que las dos formas anteriores de conseguir que los gráficos se redibujen se pueden utilizar conjuntamente, por ejemplo una en el formulario y otra en las *pictureBox*.

La Propiedad ClipControl

Por defecto esta propiedad de las *forms* y *pictureBox* está en *true*. En este caso los controles están siempre por encima de la salida de los métodos gráficos, por lo que nunca por ejemplo una línea se dibujará sobre un botón o una barra de desplazamiento.

Cuando la propiedad *ClipControl* se pone a *false* se produce una doble circunstancia:

- 1.- Los métodos gráficos situados en un evento *paint* siempre se dibujan en el plano de fondo y por tanto respetan los controles.
- 2.- Los métodos gráficos situados fuera de un evento *paint* se dibujan sobre cualquier elemento que esté en la pantalla, incluidos los controles.

2.2.4.13 Agregando Código Al Formulario Y Controlar Eventos

Se han mostrado ya las formas generales en las que se agrega código ejecutable a los procesos que controlan eventos. Es importante, sin embargo, revisar explícitamente cómo acceder al marco propio de los eventos en el cual se sitúa ese código. Esto le permitirá escoger la forma más rápida y fácil de acceder al marco de control de los eventos que necesita para agregar su código. Una vez que acceda al procedimiento adecuado para el manejo de los eventos, puede agregar código con sólo teclearlo en la ventana Código.

Es importante tener en cuenta, como hemos señalado antes, que el código relacionado con un procedimiento de control de eventos consistirá con frecuencia simplemente en llamadas a procedimientos. En este caso los procedimientos llamados contendrán el código real ejecutable. Una razón para diseñar un programa de esta forma es que el mismo procedimiento puede ser

llamado desde diferentes controladores de eventos, simplificando así, abreviando y aclarando **la arquitectura del programa**.

A continuación se indican las tres formas de "*alcanzar*" un marco de procedimiento para el manejo de eventos:

1.- Asegúrese de que esté abierta la ventana Proyecto. Seleccione el formulario al que desee agregar código de eventos. (Si está agregando un evento a un control que ha sido situado en un formulario, seleccione al formulario que se encuentre en el "*padre*" del control.)

Haga *click* en el botón Ver código. (También puede hacer *click* con el botón derecho en el icono del formulario y seleccionar Ver Código en el menú de contexto.) En el cuadro de lista Objeto, seleccione el formulario o bien otro objeto (por ejemplo, un control) al cual desee agregar código de evento. A continuación, desde el cuadro de lista Procedimiento, seleccione el controlador del procedimiento al que va a agregar código. Observe que los procedimientos controladores de código con código adjunto están en negrita en el cuadro de lista Procedimiento, mientras que los que no tienen código adjunto no lo están.

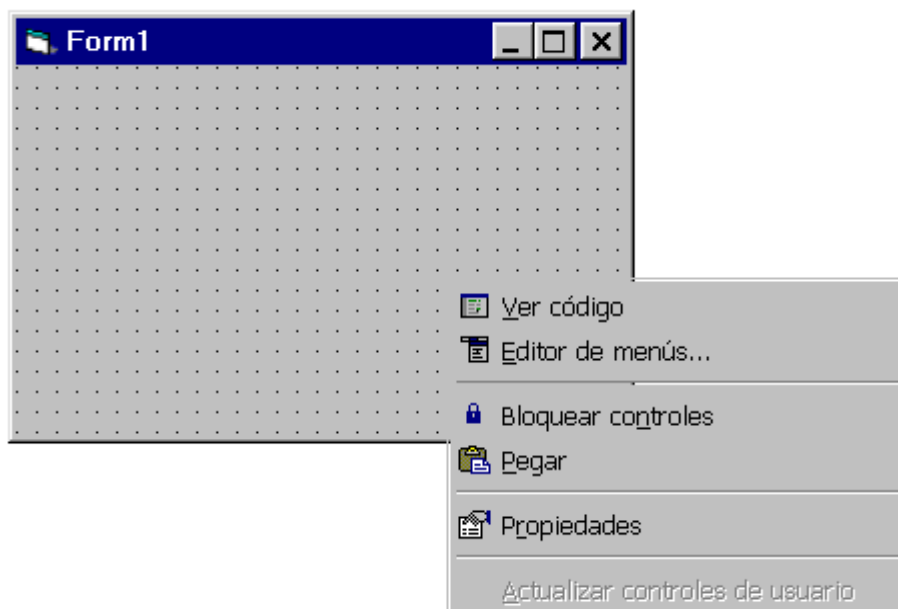


Ilustración 5 Ventana Ver Código

2.- Haga doble *click* en el formulario al cual desee agregar código. Realice su selección desde el cuadro de lista Objeto y desde el cuadro de lista Procedimiento como acabamos de describir.

3.- Haga *click* con el botón derecho en el formulario. Seleccione Ver código en el menú emergente. Realice su selección en los cuadros de la lista Objeto y Procedimiento como acabamos de describirlo.

2.2.4.14 Conclusión del estudio

La Programación Orientada a Eventos **facilita** el proceso de programación visual por su practicidad al generar rápidamente algunas aplicaciones basadas en objetos visuales. También **ayuda** al programador novato en el rápido aprendizaje de desarrollo de nuevos programas con alta calidad en muy poco tiempo.

2.2.5 La opción elegida

Visual Basic fue elegido como lenguaje de programación para la **Aplicación del Taller** por su practicidad a la hora de crear una interfaz de usuario, siendo el producto final de gran calidad sin necesitar excesivo tiempo de desarrollo.

Pero hay otros factores técnicos, tomados en cuenta al inicio del trabajo, que le hicieron idóneo para el desarrollo del proyecto. [6]

2.2.5.1 El Modelo de Aplicación de Empresarial

Este es otro de los términos que introduce Visual Studio 6.0. Se trata de la respuesta de Microsoft a la construcción de sistemas o soluciones empresariales.

He aquí algunas de las características:

☐ **A gran escala.** El sistema consiste en numerosos componentes que se despliegan físicamente sobre diversas máquinas. Tiene la capacidad de manipular enormes cantidades de datos y utilizar muchos recursos de red distribuidos. Tal sistema ha de tener, normalmente, una larga vida.

☐ **Orientado a negocio.** El sistema tiene como objeto un área de negocio o problema particular. Acompasa todos los aspectos de este problema, incluyendo la interfaz de usuario, el proceso de negocio y la manipulación de la Base de Datos.

☐ **Estratégico para la misión.** Una aplicación empresarial diseñada para satisfacer las necesidades operativas continuas de la empresa. Para ello debe ser flexible, escalable y mantenible. Además, debe proporcionar la capacidad de ser controlada en cuanto a disponibilidad y rendimiento, así como ofrecer una facilidad para la administración.

El modelo de aplicación empresarial es en realidad un conjunto de modelos que se utilizan conjuntamente para crear un sistema empresarial. Una aplicación es tan grande en alcance y complejidad que ningún individuo solo puede manipular todos los elementos que forman parte del sistema y, por lo tanto, todo el sistema está dividido en pequeños submodelos orientados a los diferentes miembros del equipo de desarrollo de la empresa.

Submodelos de la Aplicación Empresarial:

☐ **Modelo de desarrollo.** Es el modelo utilizado por los equipos de desarrollo para crear y probar la aplicación. Las prácticas y procesos de desarrollo han de estar definidos. Hay que establecer las estrategias de implantación y el control del código fuente.

☐ **Modelo de negocio.** Este modelo corresponde a los procesos, reglas, políticas y restricciones del proyecto de un negocio. Los objetivos y plazos deben estar definidos. Se consideran la seguridad, la integración en las infraestructuras existentes y el mantenimiento.

☐ **Modelo de usuario.** Trata de la interacción del usuario con el sistema. Es preciso tener en cuenta la congruencia y la integración con otras aplicaciones. Incluye asimismo la instalación, formación, documentación y configuración.

☐ **Modelo lógico.** Describe la lógica del sistema. Incluyendo la definición de objetos, servicios e interfaces.

☐ **Modelo de tecnología.** Este modelo corresponde a las tecnologías a utilizar en todos los procesos de desarrollo y en la vida de la aplicación. Es preciso seleccionar las metodologías y herramientas de desarrollo y considerar las opciones de despliegue. Asimismo a que tomar la decisión de construir, comprar o reutilizar.

☐ **Modelo físico.** Comprende el aspecto físico y la interacción del sistema. Se definen las máquinas a utilizar, así como las interconexiones de red entre componentes.

2.2.5.2 Acceso a Datos

El diseñador *Data Environment* permite gestionar visualmente conexiones y comandos de bases de datos. Proporciona toda la funcionalidad del diseñador *UserConnection* y añade la funcionalidad siguiente:

- ☐ **Conexiones múltiples.** Crea objetos de conexión múltiple en un solo entorno de datos. Cada objeto conexión puede acceder a un origen de datos diferente.
- ☐ **OLEDB.** Proporciona soporte OLEDB y ODBC.
- ☐ **Métodos de creación.** Los objetos comando que se generan en *Data Environment* se convierten en métodos utilizables en tiempo de ejecución. Un objeto comando es una definición de un tipo específico de comando que puede ejecutarse frente a un origen de datos. Es una combinación de propiedades y métodos que se puede utilizar para hacer cosas como consultar un origen de datos.
- ☐ **Colección de comandos.** Los comandos relacionados se pueden organizar en colecciones.
- ☐ **Enlace de datos.** Se pueden enlazar campos de *Data Environment* en controles de un informe o un formulario.

Crystal Reports y Data Report

El diseñador Data Report es un generador de informes que permite crear informes jerarquizados con separaciones. Usa un origen de datos, como el creado con *Data Environment*, para generar estos informes. Además de ofrecer una forma de presentar e imprimir los informes, el diseñador facilita métodos para exportar un informe a HTML o a un archivo de texto. Aunque Data Report no tiene tantas funciones como Crystal Reports, tiene mayor integración con el entorno de programación.

ADO Y OLE DB

OLE DB es una interfaz de programación de bajo nivel diseñada para acceder a diversos orígenes de datos. Puede soportar tanto orígenes de datos relacionales como no relacionales,

incluyendo mensajería, sistemas de archivos y otros orígenes de datos no tradicionales. OLE DB especifica un conjunto de interfaces COM (Modelo de objetos orientados a componentes) que se utilizan para ocultar detalles de implantación obligatorios para construir servicios de accesos a datos. La construcción de interfaces OLE DB no es una tarea sencilla, ya que requiere gran destreza en el desarrollo de COM.

Los programas escritos nunca utilizarán OLE DB directamente. Al igual que se emplea DAO (Objetos para acceso a datos) o RDO (Objetos de datos remotos) para entablar conversación con la capa ODBC (Conectividad abierta de Base de Datos), se utiliza ADO dentro de los programas para tratar con la capa OLE DB. ADO proporciona un conjunto de objetos programables que representen los datos proporcionados por el proveedor OLE DB utilizado.

ADO consiste en siete objetos, tres de los cuales son independientes y cuatro de ellos dependientes. Los independientes pueden existir por sí mismos y los dependientes necesitan una conexión con un objeto independiente.

☐ **Connection.** Este objeto representa una única sesión con un origen de datos seleccionado. Es el equivalente al tipo de conexión que hace un *control data* directamente con la Base de Datos. El objeto Connection se puede compartir con una instancia completa de una aplicación, al igual que se puede utilizar el objeto DAO database.

☐ **Command.** Este objeto especifica la definición o instrucción de manipulación de los datos a ejecutar. Puesto que ADO soporta definición y manipulación de datos, se puede crear una Base de Datos sobre la marcha.

☐ **Recordset.** Representa un conjunto de registros que devuelve un comando enviado a la Base de Datos. Se puede controlar el tipo de cursor a usar en los registros, la forma de ordenarlos y filtrarlos, así como ver las definiciones de cada campo del Recordset. Es similar al objeto DAO Recordset.

☐ **Field.** Este objeto es dependiente y existe siempre como parte de un Recordset. Al igual que con DAO, es posible recorrer la colección Fields para recuperar información sobre cada campo del Recordset. Esto es importante para crear formularios sobre la marcha, ya que simplemente hay que observar el tipo de dato y proporcionar el tipo de control correcto para ver y modificar los datos del campo.

- ☐ **Parameter.** Estos objetos son parte el objeto Command y permiten llenar valores de una consulta. Es similar al objeto *Querydef*, que también tiene una colección Parameter. El uso de objetos Parameters y Command permite pre compilar las consultas de la Base de Datos.
- ☐ **Property.** Los objetos Connection, Command, Field y Recordset contienen colecciones de objetos Property, que permiten buscar información relevante de un objeto particular.
- ☐ **Error.** El objeto Error es dependiente del objeto Connection. Cuando se produce un error durante una operación de la Base de Datos, se almacena en la colección Errors, que forma parte del objeto Connection. Se puede utilizar esa colección para determinar si se han generado errores durante la operación, lo que resulta útil si se realiza un gran número de acciones y no es preciso detenerse.

```
Public Sub Conectar()
    salir = True 'salir en caso de cerrar pantalla

    strBase = "C:\BDTallerV2.0.mdb"
    strClave = Bienvenida.clave.Text
    conexion = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source = " & strBase & ";" & " Jet OLEDB:Database Passwo
'conexion = "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;Data Source=C:\BDTallerV1.5.mdb"
'conexion = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\BDTallerV1.5.mdb"
mirecordset_cliente.CursorLocation = adUseClient
mirecordset_vehiculo.CursorLocation = adUseClient
mirecordset_vista_rapida.CursorLocation = adUseClient
mirecordset_reparacion.CursorLocation = adUseClient
mirecordset_detalle_cargos.CursorLocation = adUseClient
mirecordset_detalle_repuestos.CursorLocation = adUseClient
mirecordset_Control_Dni_Matricula.CursorLocation = adUseClient
mirecordset_Control_General.CursorLocation = adUseClient
mirecordset_cargos.CursorLocation = adUseClient
mirecordset_repuestos.CursorLocation = adUseClient
mirecordset_proveedores.CursorLocation = adUseClient
mirecordset_empleados.CursorLocation = adUseClient
mirecordset_empresa.CursorLocation = adUseClient
mirecordset_factura.CursorLocation = adUseClient
mirecordset_responsables.CursorLocation = adUseClient
```

Ilustración 6 Conexión ADO

2.2.5.3 Sinergia con otras herramientas de desarrollo

El entorno de desarrollo Visual Basic no se queda solo en Visual Basic, sino que es una *suite* de herramientas compartidas por todos los componentes de Visual Studio. El programador se encontrará a si mismo utilizando alguna de las otras herramientas de Visual Studio al enfrentarse a problemas de programación.

2.2.5.4 Programación Fiable

Los errores de Visual Basic. El lenguaje no perdona una, cualquier *cosita* que funcione mal hace que se genere un error en tiempo de ejecución el cual, sino es tratado por el código, cierra el programa. Se puede producir un error a lo largo de todo el ciclo de desarrollo, en las etapas siguientes:

- ☐ En tiempo de diseño.
- ☐ En tiempo de compilación.
- ☐ En tiempo de ejecución.

Visual Basic proporciona una serie de instrucciones integradas, métodos y objetos que se pueden utilizar para el tratamiento de errores:

- ☐ Objeto Err.
- ☐ Er1.
- ☐ Error\$.
- ☐ On Error.
- ☐ On Error Resume Next.
- ☐ Resume.

Depuración paso a paso

Visual Basic proporciona diversos métodos integrados para controlar la ejecución del programa en tiempo real. Podemos ejecutar el programa línea a línea o procedimiento a procedimiento, o bien mediante una combinación de ambos. Estas acciones de depuración básica se llaman **paso a paso**. Dado que permiten recorrer el programa, examinando las variables y la lógica, la depuración paso a paso es probablemente la herramienta de depuración más potente que ofrece Visual Basic.

En la depuración, a menudo conviene detener la ejecución en una línea determinada de código, a esto se llama **interrupción**. Visual Basic ofrece 2 tipos de puntos de interrupción: una interrupción programática que utiliza la instrucción Stop y un punto de interrupción no persistente que se define con la tecla F9.

```
Private Sub AñadirReferenciaRepuesto_LostFocus()
    Dim VRef As String
    Dim Precio As String
    Dim i As Long

    If AñadirReferenciaRepuesto.Text <> "" Then
        VRef = AñadirReferenciaRepuesto.Text
        mirecordset Control Dni Matricula.Open "select referencia, descripcion, precio, proveedor from repuestos
        If mirecordset_Control_Dni_Matricula.BOF Or mirecordset_Control_Dni_Matricula.EOF Then
            Proveedor.Text = ""
            AñadirPrecio.Text = ""
            AñadirDescripcionRepuesto = ""
            unidades.Text = ""
            'proveedor.SetFocus
        Else
            Proveedor.Text = mirecordset_Control_Dni_Matricula.Fields("proveedor")
            Precio = mirecordset_Control_Dni_Matricula.Fields("precio")
            'CAMBIAMOS COMAS POR PUNTOS
            For i = 1 To Len(Precio)
                If Mid(Precio, i, 1) = "," Then
                    Mid(Precio, i, 1) = "."
                End If
            Next
            AñadirPrecio.Text = Precio
            AñadirDescripcionRepuesto = mirecordset_Control_Dni_Matricula.Fields("descripcion")
            unidades.SetFocus
        End If
        mirecordset_Control_Dni_Matricula.Close
    End If
End Sub
```

Ilustración 7 Punto de Interrupción

Los puntos de interrupción son útiles cuando hay que ejecutar todas las secciones de código hasta cierto punto, para poder ver lo que está sucediendo. Debido a la naturaleza de Visual Basic, orientado a eventos, a menudo es preciso poder ejecutar cientos o miles de líneas de código y luego detenerse en cierta línea. Definir un punto de interrupción es la mejor forma de lograrlo.

Al utilizar un punto de interrupción, VB detiene la ejecución del programa, pero no la finaliza. Cuando se hace la pausa, se pueden examinar las variables y la estructura de los datos para localizar la causa de la lógica errónea.

2.3 Diseño Web Responsive

Hoy en día accedemos a sitios Web desde todo tipo de dispositivos; ordenador, tablet, smartphone... por lo que surge la necesidad de que las páginas Web se adapten a los diferentes tamaños y características de los mismos.

El diseño Web Responsive es una técnica de diseño Web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles.

Se trata de redimensionar y colocar los elementos de la Web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los *layouts* (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3.

El diseño Responsive permite reducir el tiempo de desarrollo, evita los contenidos duplicados, y aumenta la viralidad de los contenidos ya que permite compartirlos de una forma mucho más rápida y natural.



Ilustración 8 Diseño Responsive

2.3.1 Generaciones del diseño Web

Rápidamente el diseño Web ha ido evolucionando a medida que ha evolucionado Internet. Al principio los portales eran construidos con solo texto, pero poco a poco la inclusión de imágenes, videos y animación 3D ayudaron a caracterizar los sitios en la red, volviéndolos más interesantes e interactivos.

El diseño Web original fue mejorando y pasando por distintas generaciones, principalmente debido a mejoras en tecnología (hardware y software). Estos cambios se pueden englobar a día de hoy en cuatro generaciones, que se detallan a continuación. [7] y [8]

2.3.1.1 Primera Generación

El diseño Web de las primeras páginas era lineal y estaba orientado para científicos, que eran los usuarios que compartían su información alrededor de todo el mundo mediante estas páginas Web. La tecnología de los navegadores Web era limitada y no disponía de la suficiente capacidad de transmitir información gráfica para la correcta comunicación visual.

Las principales características de este periodo fueron:

- ☐ Las velocidades de transmisión de datos, que limitaban el peso de las páginas Web.
- ☐ El uso de monitores monocromos, que impedían la correcta visualización de la página.
- ☐ La estructuración de las páginas era bastante desordenada, con imágenes dispuestas horizontalmente y líneas de texto separadoras.

En los años 90 fue presentado al público **Mosaic**, primer navegador comercial que permitió el acceso del público a los contenidos en línea. Fue diseñado por Marc Andeesen y Eric Bina, corría originalmente en el sistema Unix.



Ilustración 9 Primer Explorador

A finales de los 80 Tim Berners-Lee propuso un proyecto basado en el hipertexto o HTML, creando por fin la primera página Web en 1991. Allí los visitantes podían consultar el proyecto, aprender sobre el hipertexto y encontrar en el portal detalles técnicos para crear una página Web. También se encontraban consejos para buscar información en la Web, ya que por entonces no existían los buscadores.

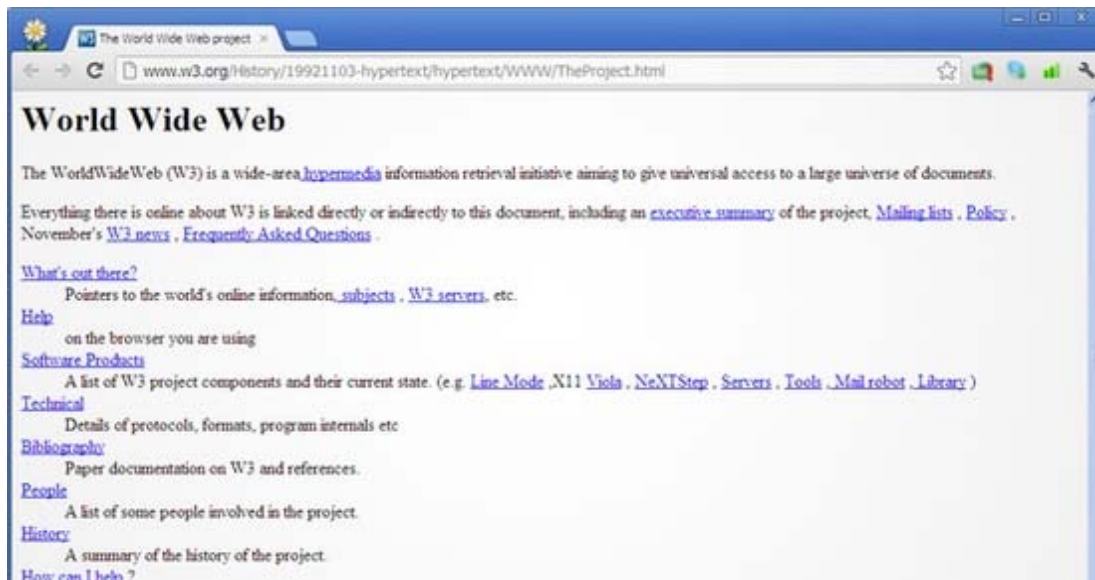


Ilustración 10 Primera Página Web 1991

En 1994, Tim Berners-Lee fundó the World Wide Web Consortium (**W3C**). La misión de este consorcio era estandarizar los protocolos y las tecnologías utilizadas para construir la Web, de manera que el contenido estuviera disponible para la mayor parte posible de la población del mundo. Se estableció HTML como el lenguaje estándar para el desarrollo de las páginas, esto generó la aparición de distintos navegadores Web.

2.3.1.2 Segunda Generación

El diseño Web de esta generación se basa en el de la etapa anterior, con la novedad de que se empieza a utilizar iconos en vez de algunas palabras.

Las principales características de este periodo fueron:

- ☐ Las páginas Web empiezan a utilizar imágenes de fondo.
- ☐ Aparece el diseño y uso de botones con relieve para facilitar la navegabilidad.

- ☐ Uso de banners en lugar de cabeceras.
- ☐ La estructuración de texto de forma jerarquizada mediante menús o listados.
- ☐ Aparición de monitores y tarjetas gráficas con mayores resoluciones y definición de color, mejorando la calidad del diseño Web.
- ☐ El uso de tablas para la organización de los contenidos y posicionamiento de los elementos, lo que aportó a los diseñadores más opciones para crear sitios Web. En un principio, sólo se utilizaban tablas para mostrar datos, pero los diseñadores supieron utilizarlas para aplicar una estructura a sus diseños.
- ☐ Se popularizan los *SpacerGIFs*, que eran imágenes transparentes utilizadas para controlar los espacios en blanco dentro de una página Web.
- ☐ Mucha diferencia en la adaptación de los estándares de los navegadores principales: Internet Explorer y Netscape Navigator.



Ilustración 11 Página Web W3C 1998

2.3.1.3 Tercera Generación

El diseño Web continúa teniendo muchas restricciones con el uso del lenguaje para los dos navegadores Web.

Las características principales de este periodo fueron:

- ☐ Dinamismo, debido a la aparición del plugin de Macromedia Flash, que revolucionaría la concepción del diseño Web.
- ☐ Se enfocan las páginas según el objetivo de las mismas: vender productos o servicios, comunidades, información, noticias.
- ☐ Navegabilidad intuitiva en las páginas para ayudar al usuario.
- ☐ La mayoría de las páginas Web que aparecen son de publicidad, venta de productos y servicios.

Al mismo tiempo que apareció flash en el mundo del diseño Web, se popularizó DHTML (*Dynamic HTML*), que consistía en una serie de tecnologías Web como JavaScript (lenguaje de *script* desarrollado para incluir scripts en páginas Web), para crear elementos animados dentro de las páginas Web.

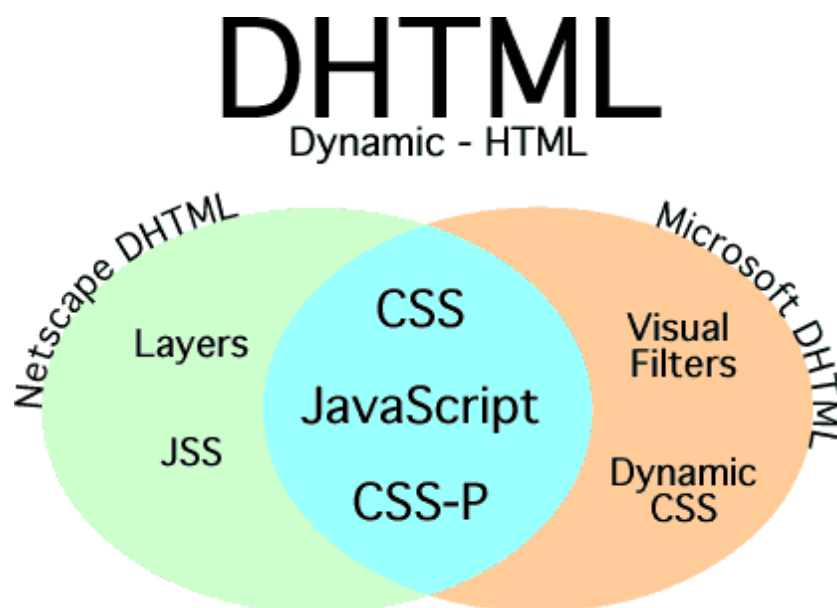


Ilustración 12 HTML dinámico

2.3.1.4 Cuarta Generación

El diseño Web en esta etapa se enfoca plenamente a los contenidos multimedia. Las características de este periodo son las siguientes:

- ☐ Se incluyen en las páginas Web elementos multimedia.
- ☐ Gran variedad de diseños gracias a las posibilidades que ofrecen las tecnologías a los programadores.
- ☐ Las últimas versiones de los navegadores soportan muchas más características y elementos en las páginas Web.

Durante este periodo comienza el auge de las hojas de estilo en cascada (Cascading Style Sheets, CSS), lenguaje que había surgido mucho antes pero que no se había utilizado debido a la incompatibilidad con algunos navegadores y el desconocimiento por parte de los diseñadores.

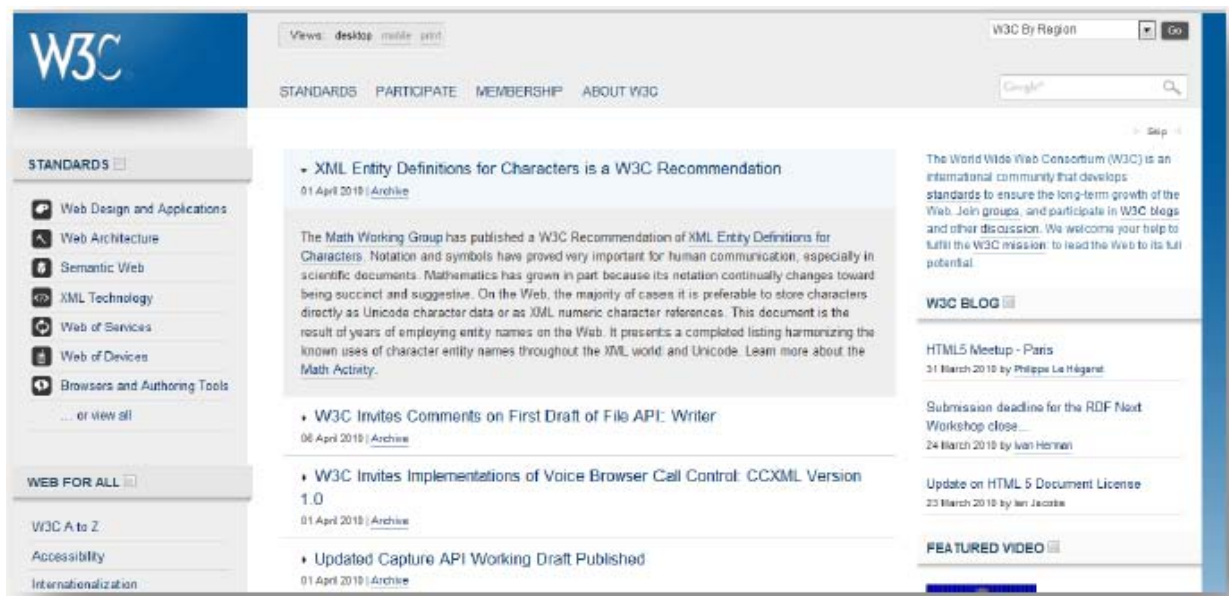


Ilustración 13 Página Web W3C 2010

El diseño Web de hoy es el diseño de cuarta generación, donde **HTML5** es la quinta revisión del lenguaje HTML y es la que se utiliza actualmente. HTML5 presenta solución a muchos problemas y errores de sintaxis en HTML4 y XHTML1.1, que fueron sus lenguajes predecesores. HTML5 intenta definir una estandarización entre la sintaxis de HTML y XHTML.

HTML5 además añade nuevas características en su sintaxis, por ejemplo, la integración de gráficas SVG y las nuevas etiquetas *video*, *audio* y *canvas*, esto hace más sencilla la integración de contenido multimedia sin ningún tipo de plugins. Entre otras nuevas etiquetas introducidas para mejorar la semántica del documento están: *header*, *article*, *section*, etc.

2.3.1.5 ¿Una Quinta Generación?

Se podría añadir una última generación de diseño Web que representaría el auge de las páginas de televisión online. La integración de televisión en la Web para empresas, ayuntamientos o asociaciones representa el movimiento más notable del diseño Web en la actualidad. [11]



Ilustración 14 Web beinsports.com

2.3.2 Estándares Web

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí. [10]

Los estándares Web son un conjunto de **Recomendaciones** dadas por la organización W3C y otras organizaciones internacionales que describen cómo crear e interpretar documentos Web. Uno de los objetivos que se persigue a la hora de crear un portal Web es que sea accesible para el mayor número de personas posible y que funcione en cualquier dispositivo de acceso a Internet.

Siguiendo las reglas de los estándares Web, el objetivo principal cuando se crea un portal Web es **separar el contenido de la apariencia** del mismo. A la hora de desarrollar una página Web hay que tener en cuenta tres elementos básicos: el contenido, la forma en la que se presenta cada elemento del contenido y su estructura. [7]

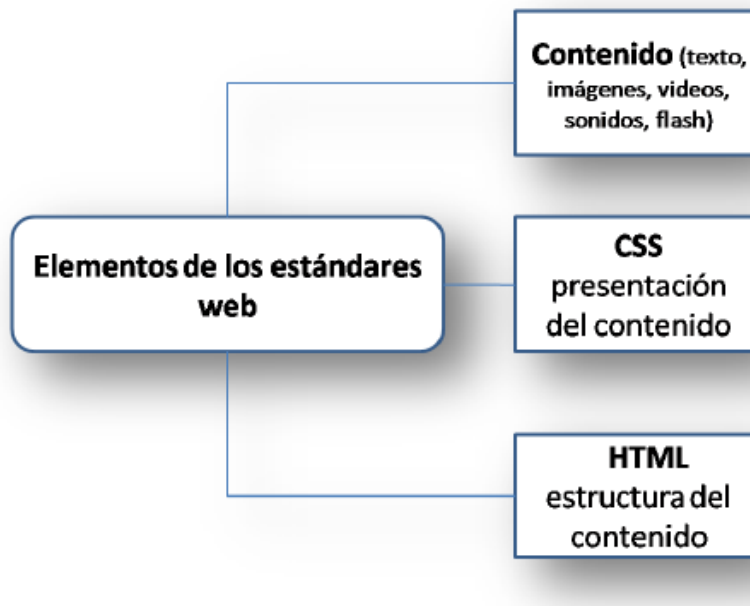


Ilustración 15 Elementos Básicos

La creación de un estándar Web requiere un proceso controlado, que consta de varias etapas que aseguran la calidad de la especificación. Este proceso permite la intervención de todos los usuarios de las tecnologías, con el objetivo de que puedan aportar su conocimiento y opiniones para la mejora de los documentos.

Tras este proceso, elaborado por especialistas en la materia, se obtienen unos estándares de calidad, y al estar disponible para todo el mundo, las especificaciones se depuran exhaustivamente antes de ser consideradas como **Recomendación**.

Estos estándares, están sujetos a la Política de Patentes del W3C, lo que permite que sean utilizados libremente por toda la comunidad Web. Al utilizar las mismas tecnologías, las máquinas se entienden entre sí y cualquier usuario puede interactuar con el resto. [10]



Ilustración 16 Distintos dispositivos y redes comunicándose a través de protocolos y lenguajes comunes

Para ayudar a los desarrolladores que deseen utilizar sus **Recomendaciones**, el W3C ofrece una serie de herramientas que permiten verificar si se hace una correcta aplicación de las especificaciones.

En Octubre de 2014, el Grupo de Trabajo de HTML publicó **HTML5** como **Recomendación** del W3C. Esta especificación define la quinta revisión importante del Hypertext Markup Language (HTML), el formato utilizado para crear páginas Web y aplicaciones, y la piedra angular de la *Open Web Platform*.

2.3.3 HTML5: Nueva especificación de HTML

HTML5 es un nuevo concepto para la construcción de sitios Web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Empecemos por el principio. Todo comenzó con una simple versión de HTML propuesta para crear la estructura básica de páginas Web, organizar su contenido y compartir información. El lenguaje y la Web misma nacieron principalmente con la intención de comunicar información por medio de texto.

El limitado objetivo de HTML motivó el desarrollo de nuevos lenguajes y programas para agregar características a la Web nunca antes implementadas. De las opciones propuestas, Java y Flash fueron las más exitosas; ambas fueron masivamente adoptadas y ampliamente consideradas como el futuro de Internet. Sin embargo, tan pronto como el número de usuarios se incrementó e Internet pasó a ser un campo estratégico para los negocios y la interacción social, las limitaciones de estas dos tecnologías se hicieron evidentes.

El mayor inconveniente de Java y Flash puede describirse como una falta de integración. Ambos fueron concebidos como complementos (plug-ins), *algo* que se inserta dentro de una estructura pero que comparte con la misma solo espacio en la pantalla.

Esta situación preparó el camino para la evolución de **Javascript**, un lenguaje interpretado incluido en navegadores. A pesar de que era evidente que Java y Flash limitaban el alcance de las aplicaciones y aislaban el contenido Web, populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la Web y solo eran ofrecidas a través de estas tecnologías. Javascript todavía no era capaz de remplazar la funcionalidad de Flash o Java.

El software para acceder a la Web continuaba evolucionando. Junto con nuevas funciones y técnicas rápidas de acceso a la red, los navegadores también mejoraron gradualmente sus intérpretes Javascript. Más potencia trajo más oportunidades y el lenguaje las aprovechó.

Javascript pasó a ser el lenguaje que permitía a los desarrolladores innovar y hacer cosas que nadie había podido hacer antes en la Web. Gracias a programadores y diseñadores Web esta tecnología superó sus limitaciones, y Javascript, HTML y **CSS** se convirtieron pronto en una combinación perfecta para la evolución de la Web.

HTML5 es una mejora de esta combinación; Propone estándares para cada aspecto de la Web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, **HTML** provee los elementos estructurales, **CSS** se encuentra concentrado en como volver esa estructura utilizable y atractiva a la vista, y **Javascript** tiene todo el poder necesario para proveer dinamismo y construir aplicaciones Web completamente funcionales.

2.3.3.1 HTML5: Mejora de la combinación

HTML5 provee básicamente tres características: **estructura, estilo y funcionalidad**; Es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5.

Pero más allá de esta integración, la estructura sigue siendo parte esencial de un documento. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la Web, la estructura se vuelve una **parte vital del documento**. Debe proveer forma, organización y flexibilidad.

Estructura del cuerpo

HTML siempre ofreció diferentes formas de construir y organizar la información dentro del cuerpo de un documento. Las tablas (**<table>**) permitían acomodar datos, texto, imágenes y herramientas dentro de las celdas. Gradualmente, otros elementos remplazaron su función permitiendo lograr lo mismo con menos código. Con el surgimiento de portales más interactivos y la integración de HTML, CSS y Javascript, el uso del elemento **<div>** se volvió una práctica común.

Pero tanto el elemento **<div>** como **<table>** no ofrecen demasiada información acerca de la parte del cuerpo que están representando. Para los usuarios estos indicios no son importantes, pero para los navegadores la correcta interpretación de qué hay dentro del documento que se está procesando es crucial en muchos casos. Con la revolución de los dispositivos móviles y el surgimiento de diferentes formas de acceso a la Web, la identificación de cada parte del documento es más relevante que nunca.

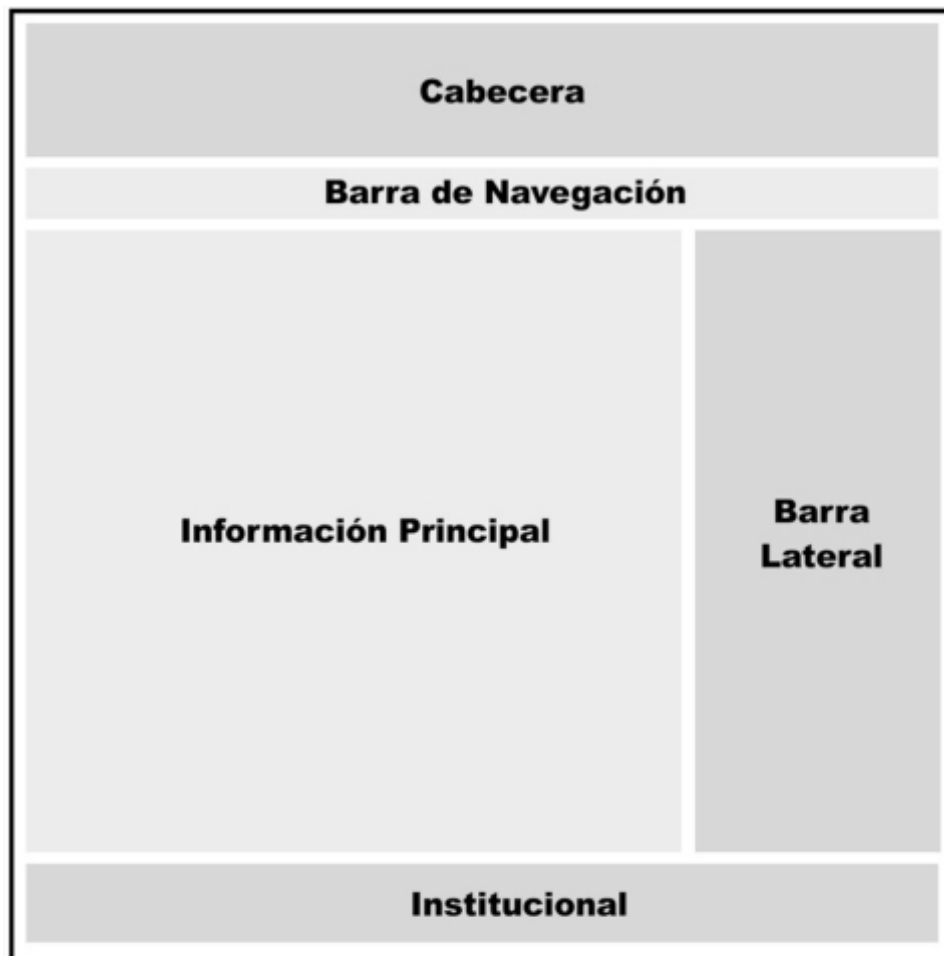


Ilustración 17 Representación visual de un clásico diseño Web

Considerando lo anterior, HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo.

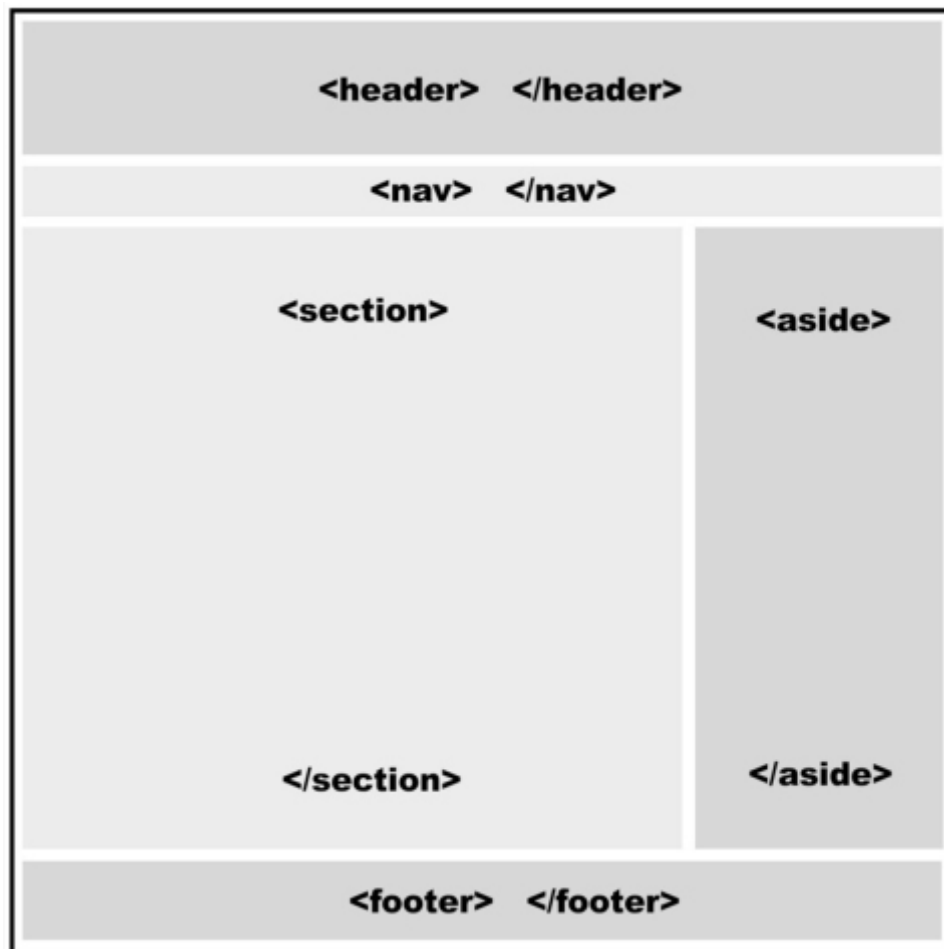


Ilustración 18 Representación visual de un diseño utilizando elementos HTML5

<header>

El elemento `<header>` no debe ser confundido con `<head>`, usado antes para construir la cabecera de documento. La intención de `<header>` es proveer información introductoria (títulos, subtítulos, logo), pero difiere respecto a `<head>` en su alcance. Mientras que el elemento `<head>` tiene el propósito de proveer información acerca de todo el documento, `<header>` es usado solo para el cuerpo o secciones específicas dentro del cuerpo.

<nav>

La barra de navegación es generada en HTML5 con el elemento `<nav>`. Se encuentra dentro de las etiquetas `<body>` pero es ubicado después de la etiqueta de cierre de cabecera `</header>`. Esto es porque `<nav>` no es parte de la cabecera sino una nueva sección.

HTML5 es versátil y sólo nos otorga los parámetros y elementos básicos con los que trabajar, pero como usarlos es decisión nuestra. El elemento `<nav>` podría ser insertado en cualquier parte del cuerpo, sin embargo, siempre se debe considerar que estas etiquetas fueron creadas para brindar información a los navegadores y ayudar a cada nuevo programa y dispositivo en el mercado a identificar las partes más relevantes del documento.

El elemento `<nav>` fue creado para ofrecer ayuda para la navegación, como en menús principales o grandes bloques de enlaces, y debería ser utilizado de esta manera.

<section>

La columna de información principal contiene la información más relevante del documento y puede ser encontrada en diferentes formas (dividida en varios bloques o columnas). El elemento en HTML5 que especifica estas secciones se llama `<section>`. La columna de información principal es una sección aparte, por este motivo va debajo de la etiqueta de cierre `</nav>`.

<aside>

La columna llamada barra lateral se ubica al lado de la información principal. Esta sección normalmente contiene datos relacionados con la información principal pero que no son igual de importantes. En HTML5 podemos diferenciar esta clase secundaria de información utilizando el elemento `<aside>`.

La etiqueta no tiene una posición definida, solo describe la información que contiene, no el lugar dentro de la estructura. Este elemento puede estar ubicado en cualquier parte del diseño y ser usado siempre y cuando su contenido no sea considerado como el contenido principal del documento.

<footer>

Para finalizar la estructura elemental de un documento HTML5 solo necesitamos un elemento más. Ya contamos con la cabecera del cuerpo, secciones con ayuda para navegación, información importante y hasta una barra lateral con datos adicionales, por lo tanto lo único que nos queda es cerrar el diseño para otorgarle un final al cuerpo del documento. HTML5 provee un elemento específico llamado `<footer>`.

Generalmente, el elemento <footer> representará el final del cuerpo de nuestro documento. Sin embargo, <footer> puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta <header>).

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>

  </section>
  <aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
  <footer>
    Derechos Reservados &copy; _____
  </footer>

</body>
</html>
```

Ilustración 19 Diseño básico estructura HTML5

Dentro del cuerpo

Los elementos HTML5 vistos anteriormente nos ayudan a identificar cada sección del diseño y asignar un propósito intrínseco a cada una de ellas, pero lo que es realmente importante se encuentra en el interior de la secciones.

Los elementos <header>,<nav>,<section>,<aside> y <footer> tienen un específico propósito estructural; ninguno de ellos declara algo acerca del contenido. La definición de contenido estará compuesta por diferentes elementos visuales como títulos, textos, imágenes, videos y

aplicaciones interactivas entre otros. Necesitamos diferenciar estos elementos y establecer una relación entre ellos dentro de la estructura.

<article>

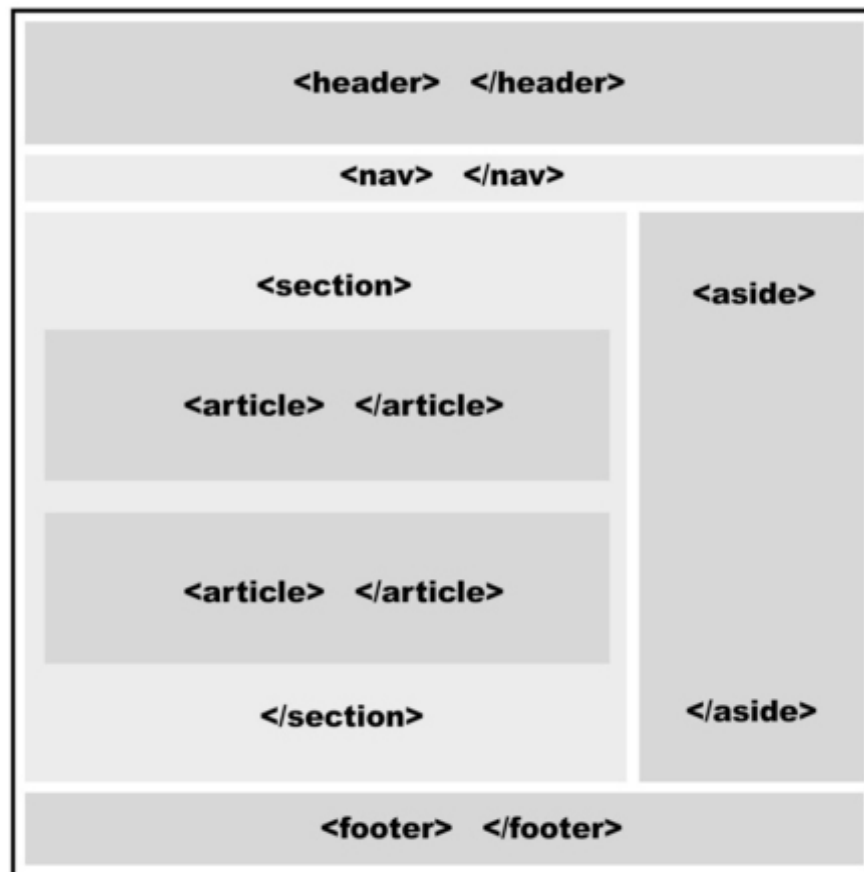


Ilustración 20 El elemento <article>

Este elemento fue creado con la intención de contener unidades independientes de contenido, por lo que puede incluir mensajes de foros, artículos de una revista digital, entradas de blog, comentarios de usuarios, etc. Agrupa porciones de información que están relacionadas entre si independientemente de su naturaleza. Como una parte independiente del documento, el contenido de cada elemento <article> tendrá su propia estructura.

<hgroup>

Este elemento es usado para agrupar elementos H cuando la cabecera tiene múltiples niveles (por ejemplo, una cabecera con título y subtítulo).

<figure>

Este elemento representa una porción independiente de contenido (imágenes, videos...) que son referenciadas desde el contenido principal.

<figcaption>

Este elemento es utilizado para mostrar una leyenda o pequeño texto relacionado con el contenido de un elemento <figure>, como la descripción de una imagen.

Los “Viejos” Elementos

HTML5 fue desarrollado con la intención de simplificar, especificar y organizar el código. Para lograr este propósito, nuevas etiquetas y atributos fueron agregados y HTML fue completamente integrado a CSS y Javascript. Estas incorporaciones y mejoras de versiones previas están relacionadas no solo con los nuevos elementos sino también con como usamos los ya existentes.

<mark>

Resalta un texto que tiene relevancia en una situación en particular o que ha sido mostrado en respuesta de la actividad del usuario.

<small>

Representa contenido al margen, como letra pequeña (restricciones legales, declaración de derechos...)

<cite>

Este elemento es utilizado para mostrar el título de un trabajo (libro, película...)

<address>

El elemento encierra información de contacto para un elemento <article> o para el documento completo. Es recomendable que sea insertado dentro de un elemento <footer>.

<time>

Se utiliza para mostrar fecha y hora en formatos comprensibles por los usuarios y navegador.

2.3.3.2 Estilos CSS

La Web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. CSS no es parte de la especificación de HTML (HTML5) y nunca lo fue. Este lenguaje es un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. CSS es implementado en paralelo, enfocado en las necesidades de los diseñadores, para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, etc.

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación HTML5 fue desarrollada considerando CSS a cargo del diseño. Debida a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo Web, razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a **CSS3**.

Las nuevas reglas: Propiedades CSS3

En un intento por reducir el uso de código Javascript y para estandarizar funciones populares, CSS3 no solo cubre el diseño y estilos Web sino también forma y movimiento. La especificación de CSS3 es presentada en módulos que permiten a la tecnología proveer una especificación estándar por cada aspecto involucrado en la presentación visual del documento. Desde esquinas redondeadas y sombras hasta transformaciones y reposicionamiento de los elementos ya presentados en pantalla.

Presentamos a continuación algunas de las nuevas propiedades que provee CSS3 para crear efectos visuales y dinámicos:

border-radius. Esta propiedad genera esquinas redondeadas para la caja formada por el elemento. Posee dos parámetros diferentes que dan forma a la esquina.

box-shadow. Esta propiedad crea sombras para la caja formada por el elemento. Puede tomar cinco parámetros: el color, el desplazamiento horizontal, el desplazamiento vertical, el valor de difuminación, y la palabra clave *inset* para generar sombra interna.

text-shadow. Propiedad similar a *box-shadow* pero específica para textos.

@font-face. Esta regla nos permite cargar y usar cualquier fuente que necesitemos.

outline. Esta propiedad fue mejorada con la incorporación de otra propiedad llamada **outline-offset**. Ambas propiedades combinadas generan un segundo borde alejado del borde original del elemento.

border-image. Esta propiedad crea un borde con una imagen personalizada. Necesita que el borde sea declarado previamente con las propiedades *border* o *border-with*.

transform. Esta propiedad modifica la forma de un elemento. Utiliza cuatro funciones básicas: *scale(escalar)*, *rotate(rotar)*, *skew(inclinar)* y *translate(mover)*.

transition. Propiedad que puede ser aplicada para crear una transición entre dos estados de un elemento.

2.3.3.3 La relevancia de Javascript

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de los motores de interpretación, creados para acelerar el procesamiento de código. Transformaron el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió confirmar a Javascript como la mejor opción para la Web.

A la vez, interfaces de programación de aplicaciones (**APIs**) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. (*Web Storage, Canvas...*)

Incorporando Javascript

Al igual que en CSS, existen tres técnicas para incorporar código Javascript dentro de HTML. Recordando que HTML5 recomienda solo la inclusión de archivos externos.

En línea. Técnica simple para insertar código Javascript en nuestro documento que se aprovecha de atributos disponibles en elementos HTML. Estos atributos son manejadores de eventos que ejecutan código de acuerdo a la acción del usuario. (*onclick, onMouseOver, onMouseOut*).

Esta clase de práctica está permitida pero no es recomendable, el código HTML se extiende innecesariamente y se hace difícil de mantener y actualizar.

```

while($fila=mysql_fetch_array ($result)){
?>
<tr id="fila_<?php echo $variable ?>" onMouseOver="ResaltarFila('fila_<?php echo $variable ?>');|"
<td width='150'><a href="zonaprivada2.php?pasa_vehiculo=<?php echo $fila['matricula']?>"><?php ech
<td width='150'><?php echo $fila['marca']; ?></td>
<td width='150'><?php echo $fila['modelo']; ?></td>
<td width='150'><?php echo $fila['matriculacion'];?></td>
</tr>

```

Ilustración 21 Ejemplo código en línea

Embebido. Para trabajar con códigos extensos y funciones personalizadas debemos agrupar los códigos en un mismo lugar entre etiquetas `<script>`. Nos ayuda a organizar el código en un solo lugar, afectando a los elementos HTML por medio de referencias.

Para mayor claridad, recomendamos siempre colocar sus códigos Javascript en la cabecera del documento y luego referenciar los elementos a ser afectados usando los métodos Javascript apropiados para el propósito.

Existen tres métodos disponibles para referenciar elementos HTML desde Javascript: **getElementsByTagName** (referencia un elemento por su nombre o palabra clave), **getElementById** (referencia por el valor de su atributo id) y **getElementsByClassName** (referencia un elemento por el valor de su atributo class).

```

20 <head>
21 <title>Zona Privada TMB</title>
22 <meta charset="utf-8">
23
24 <script language="javascript" type="text/javascript">
25 // RESALTAR LAS FILAS AL PASAR EL MOUSE
26 function ResaltarFila(id_fila) {
27     document.getElementById(id_fila).style.backgroundColor = '#C0C0C0';
28 }
29
30 // RESTABLECER EL FONDO DE LAS FILAS AL QUITAR EL FOCO
31 function RestablecerFila(id_fila) {
32     document.getElementById(id_fila).style.backgroundColor = '#FFFFFF';
33 }
34
35 </script>
36 </head>

```

Ilustración 22 Ejemplo código Embebido

Archivos Externos. Los códigos Javascript crecen exponencialmente cuando agregamos nuevas funciones y aplicamos algunas de las APIs mencionadas previamente. Los códigos embebidos incrementan el tamaño de nuestros documentos y los hacen repetitivos. Para reducir los tiempos de descarga, incrementar nuestra productividad y poder distribuir y reutilizar nuestros códigos sin comprometer eficiencia, se recomienda grabar todos los códigos Javascript en uno o más archivos externos y llamarlos usando el atributo **src**.



```
3 <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta name="description" content="Blueprint: " />
8   <meta name="keywords" content="" />
9   <meta name="author" content="Codrops" />
10  <!--<link rel="stylesheet" type="text/css" href="css3/default.css" />-->
11  <link rel="stylesheet" type="text/css" href="css/component.css" />
12  <script src="js/modernizr.custom.js"></script>
13 </head>
```

Ilustración 23 Ejemplo llamada archivo externo

3 Propuestas

3.1 Consideraciones Generales

Al ser un proyecto amplio en el tiempo podemos señalar dos grandes iteraciones de los objetivos anteriores:

1. Del papel al software de gestión.
2. De la información en local a la Web.

En ambos casos se comenzará realizando un estudio de los sistemas presentes en TMB para conocer cuáles son sus principales funcionalidades, qué ventajas aportan y qué inconvenientes presentan en cuanto a la organización y gestión de la información. Este estudio será la base para el análisis de requisitos del sistema.

Tras este estudio, se elegirá una nueva tecnología tanto para el de desarrollo del software como para el sitio Web, de forma que se solventen los problemas identificados en el análisis.

Para finalizar, diseñamos y desarrollamos la solución informática atendiendo a los distintos manuales y tutoriales.

Medios necesarios para el desarrollo del proyecto

Para la realización del proyecto, serán necesarios distintos medios de diversa naturaleza, hardware y software. Respecto a las necesidades de hardware, destacamos el servidor físico donde se aloja el portal web. A las máquinas del proveedor de servicios hay que sumar el ordenador personal del proyectante, que será utilizado durante todas las fases del proyecto.

En lo referente al software necesario, destaca **XAMPP**, software libre, que consiste principalmente en el sistema de gestión de bases de datos **MySQL**, el servidor Web Apache y los intérpretes para lenguajes de script: **PHP** y Perl. Gracias a XAMPP generamos los archivos necesarios en **HTML5**, JavaScript y CSS.

También hay que hacer mención a la utilización de lenguajes de programación orientado a objetos sobre **Access** para la plataforma de gestión en local. No hay que olvidar los sistemas

operativos de las máquinas mencionadas anteriormente, distintos navegadores Web, herramientas destinadas al procesamiento de textos y editores.

3.2 Software de gestión TMB

Tras estudiar la solicitud de la empresa realizaremos el Análisis, Diseño e Implementación del sistema de información, Software de Gestión de escritorio, siguiendo la metodología MÉTRICA Versión 3.

3.2.1 La Solicitud

TMB, taller de reparaciones de vehículos, necesita informatizar su sistema de gestión de clientes. Almacena y trabaja con papel todo lo referente al tratamiento de clientes, proveedores, trabajadores. La facturación, las órdenes de reparación y la gestión del almacén se realizan de forma manual.

Solicita la informatización de su sistema de información. Quiere tener un histórico de las reparaciones de los vehículos, gestionar el almacén y a los proveedores de forma eficiente. Necesita minimizar el tiempo a la hora de consultar los datos técnicos de las reparaciones y creación de hojas de trabajo.

En definitiva quiere gestionar de forma eficiente todos los aspectos del trabajo de un taller de automoción a través de una aplicación de escritorio. El sistema será responsabilidad del personal de oficina, debe ser un sistema de fácil manejo con recursos de software y hardware mínimos.

3.2.2 Análisis del sistema de información

Mediante este proceso se conseguirá la especificación detallada del Sistema de Información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema. Los resultados serán el punto de partida del Diseño del Sistema de Información.

3.2.2.1 Definición del Sistema

Vamos a definir el alcance del sistema y una visión aproximada del entorno tecnológico sobre el que se debe apoyar.

El desarrollo de la aplicación afectará a la estructura organizativa de la empresa, ya que actualmente son los mecánicos quienes realizan las hojas de trabajo por su dificultad técnica a la hora de calcular los tiempos y determinar que trabajos se deben realizar. El nuevo software traerá consigo una nueva figura, el administrativo, que será capaz de realizar las tareas técnicas apoyándose en el software de gestión y distintos programas de valoración de trabajos.

La empresa dispone de una suite ofimática (**Microsoft Office**) sobre la que apoyar el software de escritorio. Utilizaremos Access como sistema de gestión para la Base de Datos, en ella almacenaremos los datos de los clientes, vehículos, reparaciones, proveedores, almacén, trabajadores y facturación.

Partimos de un requisito de hardware y software mínimos, por lo que montaremos el software directamente sobre **Access**, utilizando un lenguaje dirigido por eventos para hacer la interfaz de usuario lo más sencilla e intuitiva posible (**Visual Basic**). En la oficina se instalará un único equipo capaz de soportar la aplicación.

En la aplicación de escritorio podemos distinguir varias zonas (**5**):

Pantalla de inicio, donde mediante una búsqueda sencilla sea posible visualizar la información más importante del cliente y sus vehículos.

Gestión de la Base de Datos, en la que se realizarán todo tipo de actualizaciones de los datos de los clientes, vehículos, almacén, proveedores y trabajadores.

Libro de Mantenimiento, un espacio donde consultar el histórico de un vehículo.

Ordenes de Reparación, la consola donde se preparará la hoja de trabajo que el mecánico debe seguir. En ella se detallan también los repuestos que debe utilizar en las distintas tareas.

Facturación, zona de transformación de la hoja de reparación en factura.

3.2.2.2 La Situación Actual

Con el estudio de la situación actual vamos a decidir el modo en que los usuarios van a acceder al sistema y determinaremos la arquitectura general que se va a implantar.

La empresa gestiona toda la información en papel (clientes, proveedores, facturación), solo posee un equipo de bajo rendimiento para visualizar datos técnicos de los coches y para consultar una guía de tiempos de reparación.

No hay arquitectura de red. Al estar toda la información recogida en papel nos encontramos con inconsistencia de datos, siendo este el gran problema junto con la pérdida de tiempo por parte de personal cualificado, ya que realizan las órdenes de reparación y facturas por su complejidad técnica.

No existe ningún mecanismo de protección de datos para evitar la fuga de información sensible de los clientes, por lo que la aplicación futura requerirá autenticación al administrativo para adecuarlo al Documento de Seguridad de la empresa.

3.2.2.3 Definición de los Requisitos

Analizando el entorno de trabajo actual observamos unas necesidades básicas a satisfacer.

Descripción de las necesidades básicas
Facilidad de manejo y gestión
Seguridad en la operaciones y protección de datos
Informatizar la gestión de papel
Recursos software y hardware mínimos

Facilidad de manejo y gestión

- La aplicación va a ser gestionada por un administrativo sin conocimientos técnicos de mecánica, por lo que la interfaz debe ser manejable e intuitiva en los aspectos relativos a las reparaciones y sencilla en su entorno informático.

Seguridad en las operaciones y protección de datos

- Los usuarios del sistema deben tener los permisos de acceso de TMB, al arrancar el sistema debe aparecer una petición de identificación.
- La aplicación trabaja off-line, existe un riesgo mínimo ante ataques de usuarios ajenos al sistema. Con un antivirus general será suficiente.

Informatizar la gestión de papel

- Existe una necesidad crítica de informatizar la gestión que ahora se realiza en papel, evitaremos inconsistencias en los datos así como liberar al personal cualificado de trabajos que no deberían realizar.

3.2.2.4 Catalogación de Requisitos

Basándonos en las operaciones que debe realizar el sistema podemos catalogar los requisitos de la aplicación de la siguiente forma:

Requisitos Funcionales

Descripción	Prioridad
El Usuario debe acceder al sistema mediante claves (login y password), nada más arrancar la aplicación.	↑↑
Mediante una búsqueda sencilla el usuario debe visualizar los datos técnicos del vehículo, los personales del propietario y los datos más relevantes del mantenimiento del vehículo. La búsqueda se realizará por Matrícula, DNI, Nombre y Apellido 1.	↑↑
El Usuario será el responsable del mantenimiento de los datos del cliente. Puede Crear, Borrar o Modificar los datos de los clientes en la pantalla de gestión de BBDD.	↑↑
El Usuario será el responsable del mantenimiento de los datos del vehículo. Puede Crear, Borrar o Modificar los datos de los vehículos en la pantalla de gestión de BBDD.	↑↑
El Usuario será el responsable del mantenimiento de los datos del proveedor. Puede Crear, Borrar o Modificar los datos de los proveedores en la pantalla de gestión de BBDD.	↑
El Usuario será el responsable del mantenimiento de los datos del almacén. Puede Crear, Borrar o Modificar los datos de los repuestos en la pantalla de gestión de BBDD.	↑
El Usuario será el responsable del mantenimiento de los datos de los trabajos a realizar. Puede Crear, Borrar o Modificar los datos de los cargos varios en la pantalla de gestión de BBDD.	↑
El Usuario será el responsable del mantenimiento de los datos corporativos de	↑↑

la empresa, también en la pantalla de gestión de BBDD.	
El Usuario será el responsable del mantenimiento de los datos de los empleados de TMB. Puede Crear, Borrar o Modificar los datos de los trabajadores en la pantalla de gestión de BBDD.	↑
El Usuario utilizará la pantalla de Libro de Mantenimiento para Consultar, Crear, Eliminar o Modificar una reparación de un vehículo dado por búsqueda.	↑↑
El Usuario facturará a partir de una hoja de reparación cerrada en la pantalla de Facturación.	↑↑
El Usuario debe tener la posibilidad de cerrar sus sesión	↑↑
El Usuario tendrá la posibilidad de imprimir una Orden de Reparación	↑
El Usuario tendrá la posibilidad de imprimir un Resguardo de Depósito	↑
El Usuario tendrá la posibilidad de imprimir una Factura	↑

Requisitos de Implantación

RECURSOS SOFTWARE Y HARDWARE MINIMOS

La aplicación será gestionada por un administrativo con conocimientos de ofimática, pero aun así no debe implicar amplios conocimientos de informática para facilitar su utilización.

El software debe ser capaz de ejecutarse en un equipo de nivel medio para satisfacer los requisitos de hardware mínimos. Será implementado aprovechando la suite ofimática de la empresa (**Access**) bajo el entorno de programación **Visual Studio**.

DEFINICION DEL ENTORNO TECNOLÓGICO DEL SISTEMA

El sistema estará compuesto por una Base de Datos en local (**Access**), sobre la que se programará directamente en **Visual Basic** para su gestión. Todo el trabajo se efectuará off-line, siendo sólo necesario un equipo de media potencia para “correr” el programa de escritorio.

Requisitos de Disponibilidad

La aplicación debe estar siempre disponible para su ejecución, pero con toda seguridad sólo estará activa en horario laboral.

Requisitos de Rendimiento

La elección más importante estará en la adquisición de un equipo *decente* para el funcionamiento de la aplicación. Ni Access ni Visual Basic consumen excesivos recursos.

Requisitos de Seguridad

La empresa dispone de documento de seguridad, por lo que los puntos relativos a copias de seguridad, respaldos y errores están cubiertos. Pero aun así la aplicación debe ser cuidadosa a la hora de permitir el acceso al administrador registrado. (Login y Password)

3.2.2.5 Casos de Uso

En el caso de orientación a objetos se especifican, además, los casos de uso asociados a los requisitos funcionales.

Los casos de uso son una técnica de especificación de requisitos válida tanto en desarrollos estructurados como en orientación a objetos, aunque en este último caso se propone como técnica obligatoria al ser necesaria como referencia a lo largo de todo el ciclo de vida.

Vamos a identificar el actor (único en esta ocasión), casos de uso y realizaremos una breve explicación de cada caso.

- **Administrativo.** Será el encargado de toda la aplicación. Gestionará la Base de Datos y preparará las órdenes de trabajo que posteriormente facturará.

A continuación, se muestran los **Diagramas de Casos de Uso** que representan las principales funcionalidades que el sistema deberá tener.

Descripción Gráfica

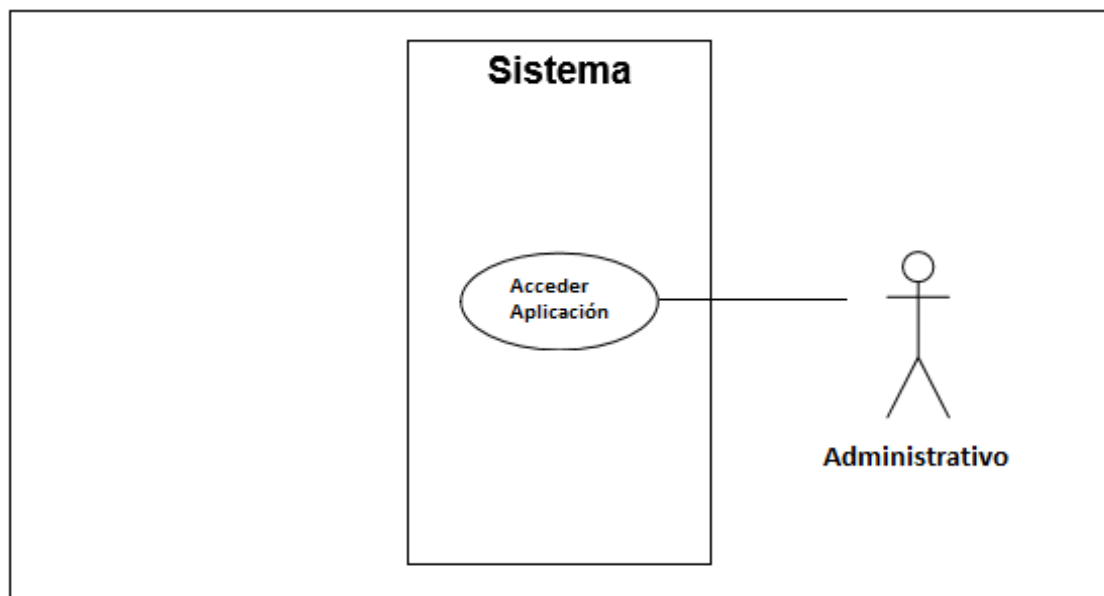


Ilustración 24. Nivel 0 - Diagrama de Casos de Uso

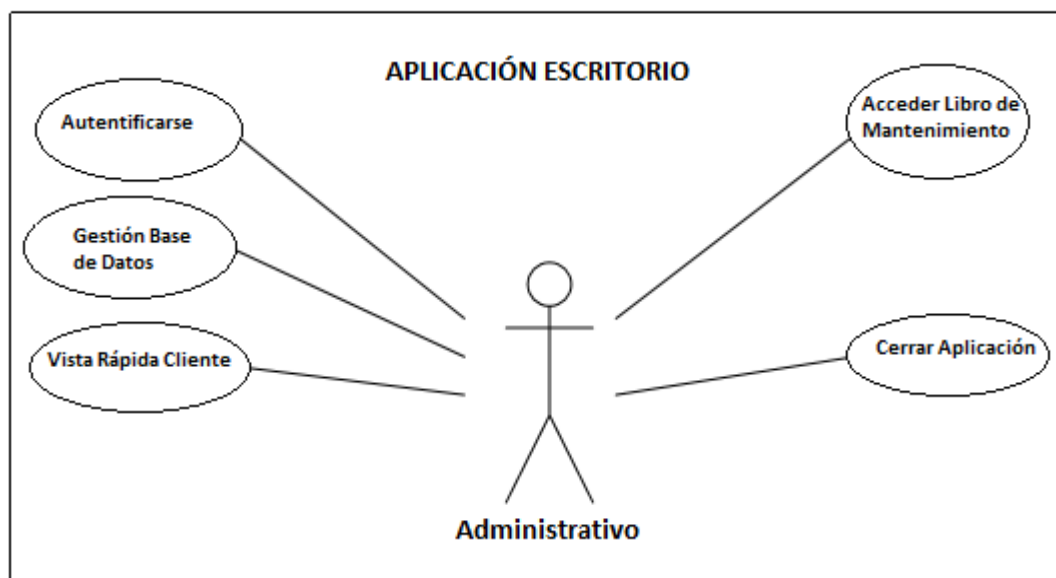


Ilustración 25. Nivel 1 - Diagrama de Casos de Uso "Aplicación Escritorio"

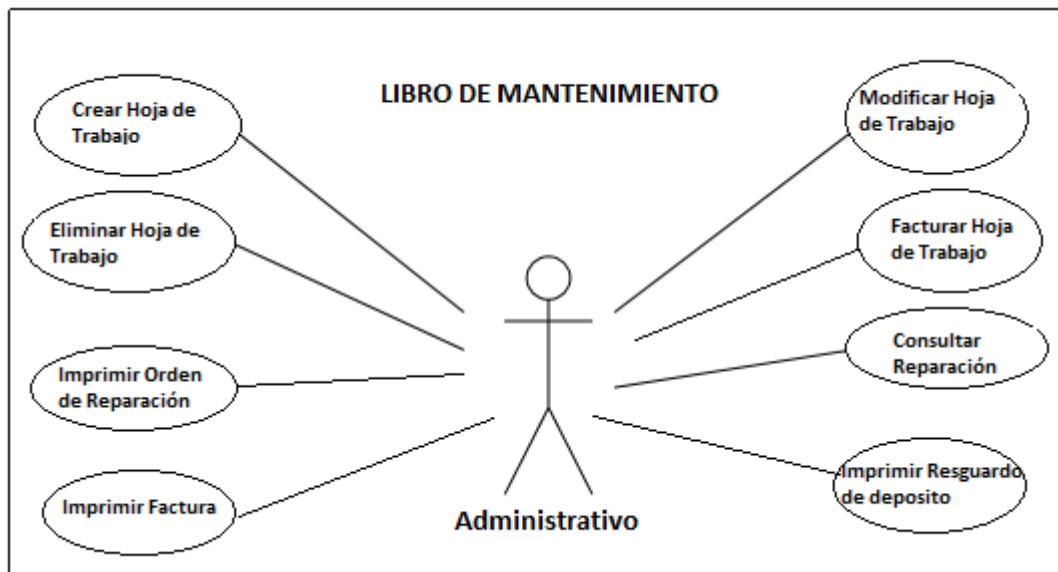


Ilustración 26. Nivel 2 - Diagrama de Casos de Uso "Libro de Mantenimiento"

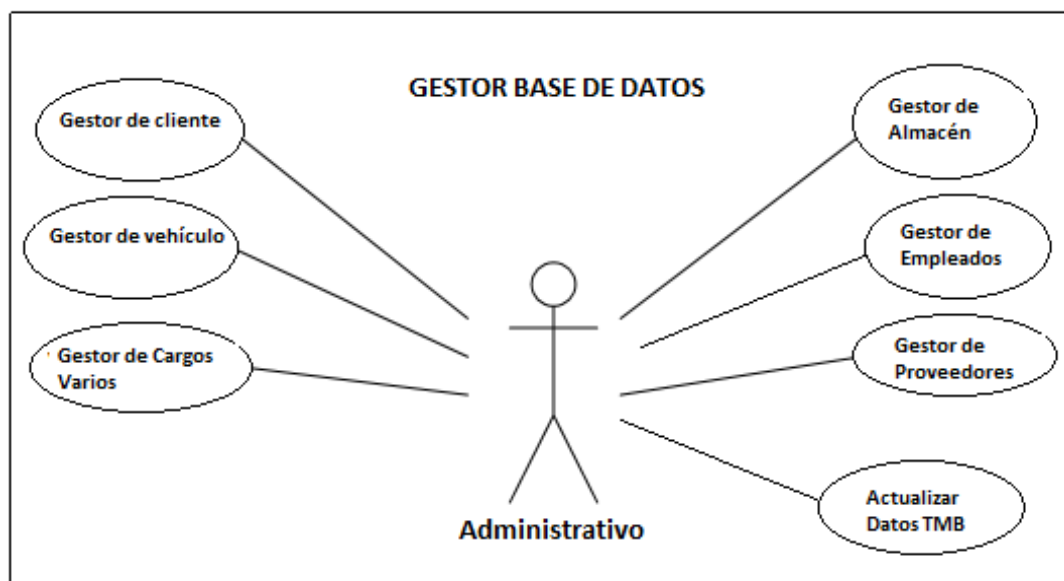


Ilustración 27. Nivel 2 - Diagrama de Casos de Uso "Gestor de bases de datos"

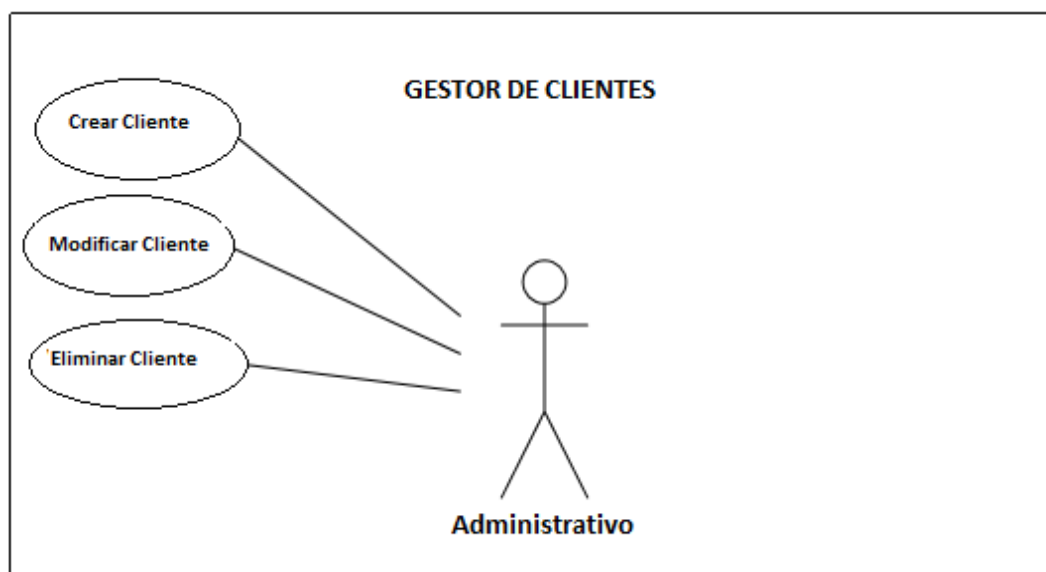


Ilustración 28. Nivel 3 - Diagrama de Casos de Usos "Gestor de Clientes"

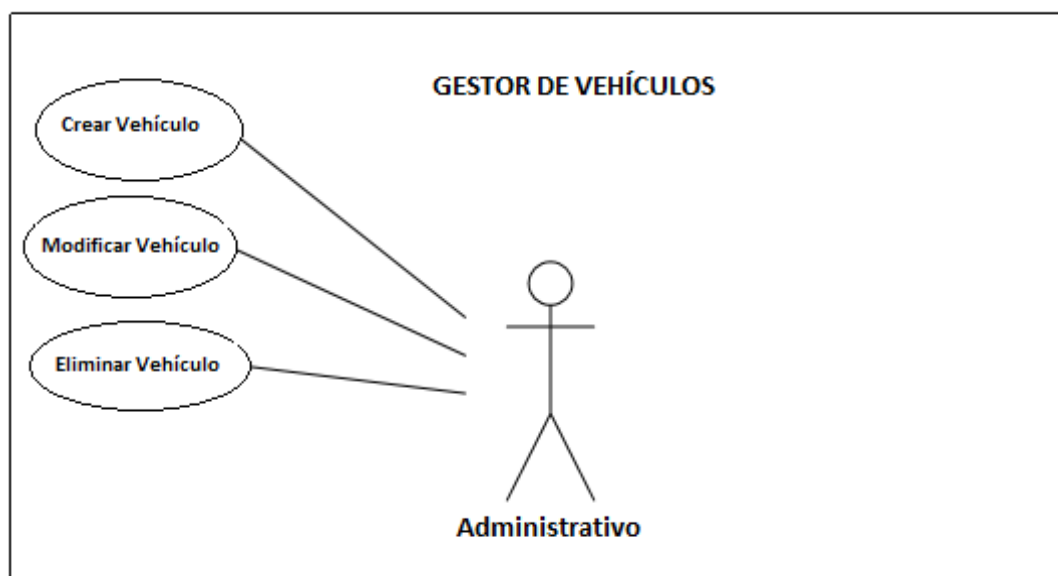


Ilustración 29. Nivel 3 - Diagrama de Casos de Usos "Gestor de Vehículos"

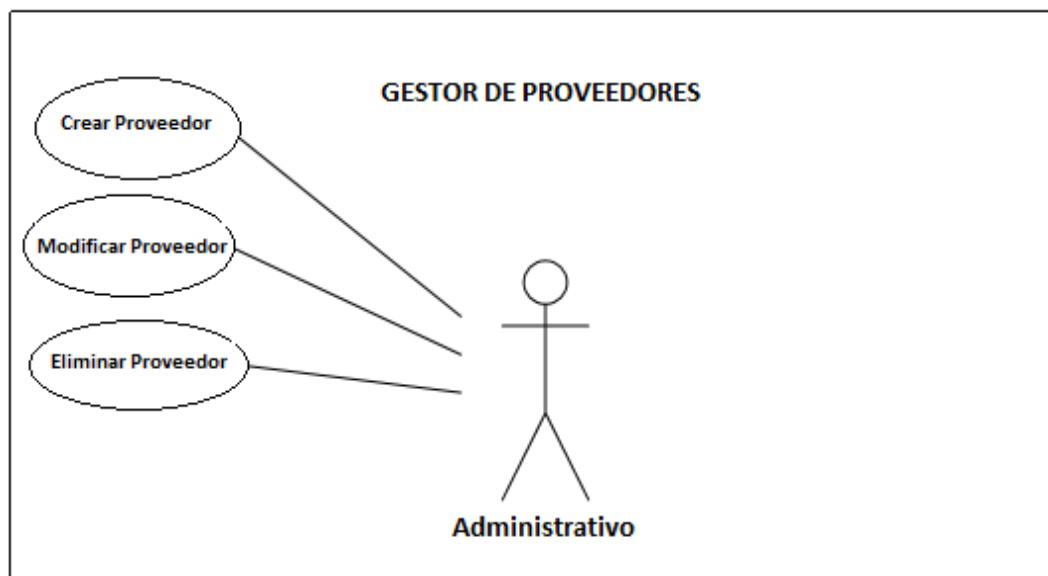


Ilustración 30. Nivel 3 - Diagrama de Casos de Usos "Gestor de Proveedores"

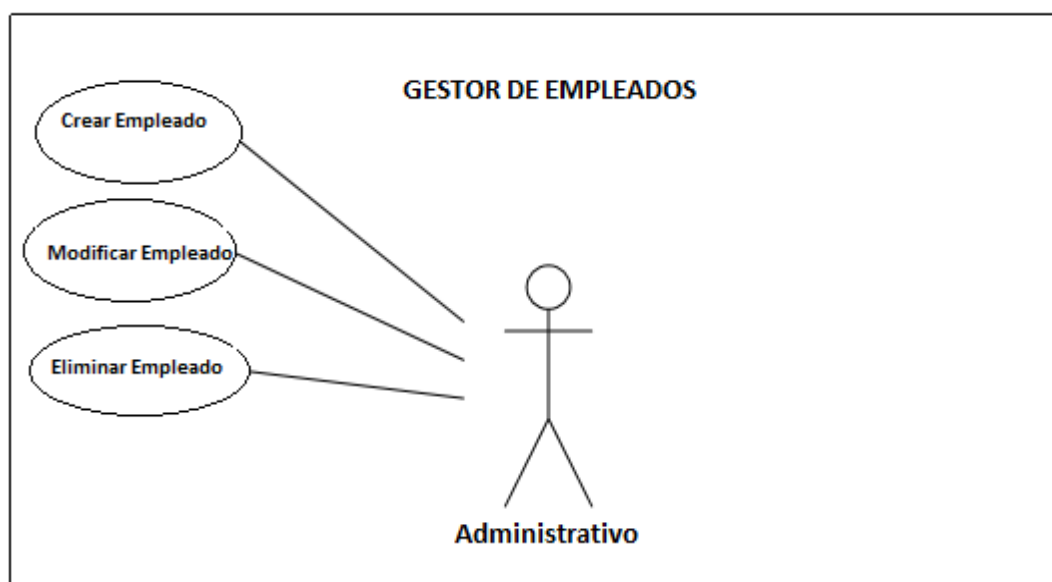


Ilustración 31. Nivel 3 - Diagrama de Casos de Usos "Gestor de Empleados"

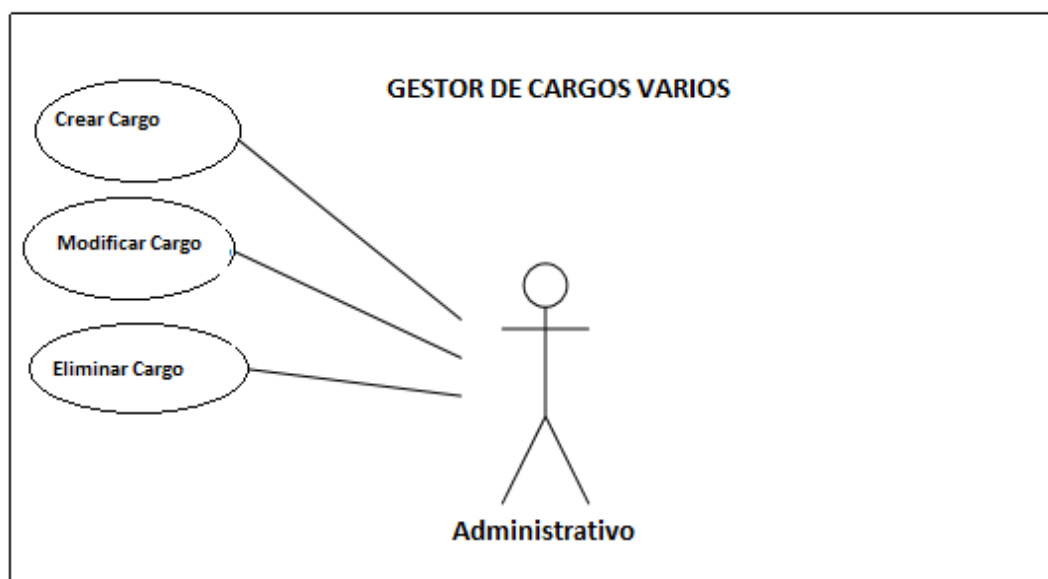


Ilustración 32. Nivel 3 - Diagrama de Casos de Usos "Gestor de Cargos Varios"

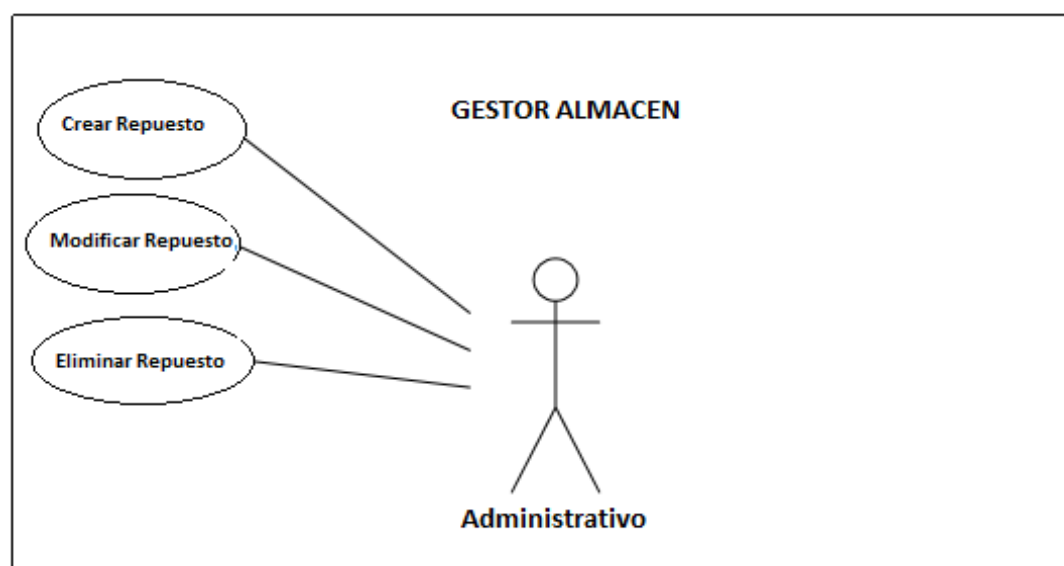


Ilustración 33. Nivel 3 - Diagrama de Casos de Usos "Gestor de Almacén"

Descripción Textual

Caso de Uso	Autenticarse
Actor	Administrativo
Objetivo	Conseguir acceso a la aplicación
Descripción	Nada más ejecutar la aplicación se pedirá claves de acceso para empezar a trabajar.

Caso de Uso	Cerrar Aplicación
Actor	Administrativo
Objetivo	Cerrar la aplicación de forma segura
Descripción	El usuario debe cerrar el software de forma segura. Terminando también con la conexión de la Base de Datos.

Caso de Uso	Vista Rápida Cliente
Actor	Administrativo
Objetivo	Visualizar los datos más importantes del cliente y sus vehículos.
Descripción	El usuario mediante una búsqueda sencilla puede visualizar datos personales del cliente y la información más relevante de sus vehículos en cuanto a mantenimiento.

Caso de Uso	Crear Hoja de Trabajo
Actor	Administrativo
Objetivo	Crear una orden de reparación nueva
Descripción	El usuario crea una nueva hoja de trabajo desde el libro de mantenimiento del vehículo seleccionado.

Caso de Uso	Modificar Hoja de Trabajo
Actor	Administrativo
Objetivo	Modificar una orden de reparación
Descripción	El usuario puede modificar una hoja de trabajo desde el libro de mantenimiento del vehículo seleccionado.

Caso de Uso	Eliminar Hoja de Trabajo
Actor	Administrativo
Objetivo	Borrar una orden de reparación
Descripción	El usuario puede borrar una hoja de trabajo desde el libro de mantenimiento del vehículo seleccionado.

Caso de Uso	Consultar Reparación
Actor	Administrativo
Objetivo	Visualizar una orden de reparación
Descripción	El usuario puede consultar una hoja de trabajo desde el libro de mantenimiento del vehículo seleccionado.

Caso de Uso	Facturar Hoja de Trabajo
Actor	Administrativo
Objetivo	Facturar una orden de reparación
Descripción	El usuario puede facturar una hoja de trabajo ya cerrada desde la consola de facturación.

Caso de Uso	Imprimir Factura
Actor	Administrativo
Objetivo	Imprimir Factura
Descripción	El usuario tiene la posibilidad de imprimir la Factura de la Reparación.

Caso de Uso	Imprimir Orden de Reparación
Actor	Administrativo
Objetivo	Imprimir Hoja de Trabajo
Descripción	El usuario tiene la posibilidad de imprimir la Orden de Reparación para el mecánico.

Caso de Uso	Imprimir Resguardo de Depósito
Actor	Administrativo
Objetivo	Imprimir Resguardo de Depósito
Descripción	El usuario tiene la posibilidad de imprimir el Resguardo de Depósito para el cliente.

Caso de Uso	Actualizar Datos de TMB
Actor	Administrativo
Objetivo	Modificar datos corporativos de la empresa
Descripción	El usuario puede actualizar la información corporativa de la empresa desde el gestor de la BD.

Caso de Uso	Crear Cliente
Actor	Administrativo
Objetivo	Crear un nuevo cliente
Descripción	El usuario puede crear un cliente desde el gestor de la BD.

Caso de Uso	Modificar Cliente
Actor	Administrativo
Objetivo	Modificar la información de un cliente
Descripción	El usuario puede actualizar los datos de un cliente desde el gestor de la BD.

Caso de Uso	Eliminar Cliente
Actor	Administrativo
Objetivo	Borrar un cliente
Descripción	El usuario puede borrar un cliente desde el gestor de la BD.

Caso de Uso	Crear Vehículo
Actor	Administrativo
Objetivo	Crear un nuevo vehículo
Descripción	El usuario puede crear un vehículo desde el gestor de la BD.

Caso de Uso	Modificar Vehículo
Actor	Administrativo
Objetivo	Actualizar la información de un vehículo
Descripción	El usuario puede modificar los datos de un vehículo desde el gestor de la BD.

Caso de Uso	Eliminar Vehículo
Actor	Administrativo
Objetivo	Borrar un vehículo
Descripción	El usuario puede borrar un vehículo desde el gestor de la BD.

Caso de Uso	Crear Proveedor
Actor	Administrativo
Objetivo	Crear un nuevo proveedor
Descripción	El usuario tiene la posibilidad de dar de alta a un proveedor desde el gestor de la BD.

Caso de Uso	Modificar Proveedor
Actor	Administrativo
Objetivo	Actualizar la información de un proveedor.
Descripción	El usuario puede modificar los datos de un proveedor desde el gestor de la BD.

Caso de Uso	Eliminar Proveedor
Actor	Administrativo
Objetivo	Borrar un proveedor
Descripción	El usuario tiene la posibilidad de dar de baja a un proveedor desde el gestor de la BD.

Caso de Uso	Crear Empleado
Actor	Administrativo
Objetivo	Crear un nuevo empleado
Descripción	El usuario tiene la posibilidad de dar de alta a un empleado desde el gestor de la BD.

Caso de Uso	Modificar Empleado
Actor	Administrativo
Objetivo	Actualizar la información de un empleado
Descripción	El usuario puede modificar los datos de un empleado desde el gestor de la BD.

Caso de Uso	Eliminar Empleado
Actor	Administrativo
Objetivo	Borrar un empleado
Descripción	El usuario tiene la posibilidad de dar de baja a un empleado desde el gestor de la BD.

Caso de Uso	Crear Cargo Vario
Actor	Administrativo
Objetivo	Crear un nuevo concepto de reparación
Descripción	El usuario tiene la posibilidad de dar de alta un nuevo cargo desde el gestor de la BD.

Caso de Uso	Modificar Cargo Vario
Actor	Administrativo
Objetivo	Actualizar un concepto de reparación
Descripción	El usuario tiene la posibilidad de modificar la información de un cargo desde el gestor de la BD.

Caso de Uso	Eliminar Cargo Vario
Actor	Administrativo
Objetivo	Borrar un concepto de reparación
Descripción	El usuario tiene la posibilidad de eliminar un cargo desde el gestor de la BD.

Caso de Uso	Crear Repuesto
Actor	Administrativo
Objetivo	Crear un nuevo repuesto
Descripción	El usuario tiene la posibilidad de dar de alta un nuevo repuesto desde el gestor de la BD.

Caso de Uso	Modificar Repuesto
Actor	Administrativo
Objetivo	Actualizar un repuesto
Descripción	El usuario tiene la posibilidad de modificar la información de un repuesto desde el gestor de la BD.

Caso de Uso	Eliminar Repuesto
Actor	Administrativo
Objetivo	Borrar un repuesto
Descripción	El usuario tiene la posibilidad de dar de baja un repuesto desde el gestor de la BD.

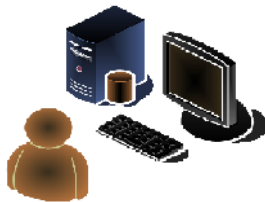
3.2.3 Diseño del Sistema de Información

Una vez efectuada la propuesta y el posterior análisis de requisitos y casos de uso, se procede a realizar el diseño de la Aplicación de Escritorio.

3.2.3.1 Diseño Arquitectónico

Microsoft Office ofrece un Sistema Gestor de Bases de Datos (Access), con el que vamos a crear nuestra Base de Datos. La Base de Datos del Taller será almacenada en un Equipo que trabajará

fuera de Red y será gestionada por un único tipo de usuario (Personal de Oficina), que a su vez será el Administrador de toda la aplicación.



El equipo informático debe ser lo bastante potente para que la **Aplicación de Escritorio**, que gestiona nuestra Base de Datos, funcione con regularidad. Ni Access ni la aplicación en Visual Basic necesitan excesivos recursos de rendimiento.

3.2.3.2 Diseño de la Base de Datos

Nuestra Base de Datos **relacional** almacenará todos los datos relativos al funcionamiento de un taller de reparación de vehículos, información personal de los clientes, datos técnicos de sus vehículos, gestión de almacén, información de proveedores, empleados, trabajos realizados...

Un buen diseño de la Base de Datos es **crucial** para que la aplicación de escritorio funcione adecuadamente. El diseño debe permitir que a partir de un cliente de dado, accedamos a todos los vehículos de su propiedad, seleccionar uno de ellos y consultar todas sus reparaciones, que a su vez deben reflejar los repuestos utilizados de nuestros proveedores.

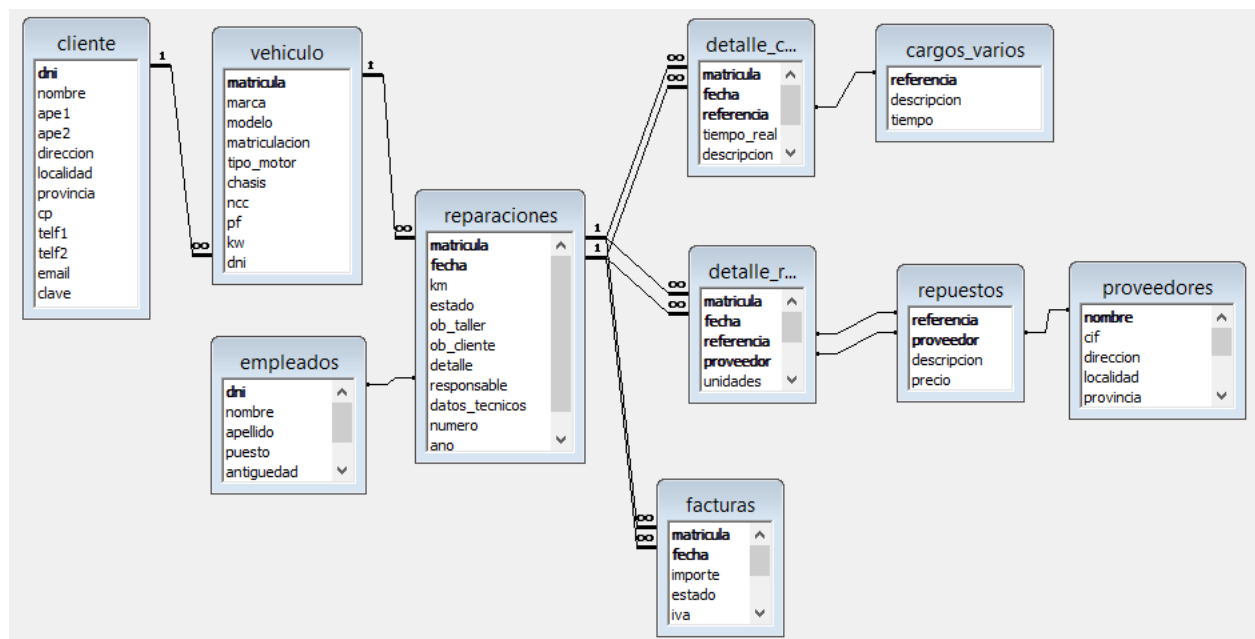


Ilustración 34. Diseño relacional Base de Datos

La Base de Datos será mantenida en su totalidad por la Aplicación de Escritorio, la simbiosis entre el código del software y las relaciones de la Base de Datos debe ser perfecta para evitar inconsistencias de datos.

Descripción de las tablas

Tabla tmb. Recoge la información corporativa de la empresa, datos de contacto, precio de la Hora de Trabajo e IVA.

	Nombre del campo	Tipo de datos
🔑	dif	Texto
	nombre	Texto
	direccion	Texto
	provincia	Texto
	localidad	Texto
	cp	Número
	telf1	Texto
	telf2	Texto
	email	Texto
	fax	Texto
	n_registro	Texto
	precio	Moneda
	iva	Moneda

Tabla cliente. Almacena los datos personales del cliente, destacando su clave privada.


	Nombre del campo	Tipo de datos
	dni	Texto
	nombre	Texto
	ape1	Texto
	ape2	Texto
	direccion	Texto
	localidad	Texto
	provincia	Texto
	cp	Número
	telf1	Texto
	telf2	Texto
	email	Texto
	clave	Texto

Tabla vehiculo. Recoge todos los datos técnicos del vehículo, las referencias de mantenimiento más importantes y las fechas de ITV, fecha reservada y próxima revisión. Un cliente puede tener varios vehículos.

	Nombre del campo	Tipo de datos
	matricula	Texto
	marca	Texto
	modelo	Texto
	matriculacion	Fecha/Hora
	tipo_motor	Texto
	chasis	Texto
	ncc	Texto
	pf	Moneda
	kw	Moneda
	dni	Texto
	prox_revision	Número
	fecha_reservada	Fecha/Hora
	prox_itv	Fecha/Hora
	distribucion	Número
	f_aire	Texto
	f_aceite	Texto
	f_habitaculo	Texto
	f_combustible	Texto
	aceite	Texto
	neumaticos	Texto
	intervalo_distribucion	Texto
	ref_distribucion	Texto
	cod_radio	Texto
	cod_llave	Texto
	aceitecc	Texto
	axone	Texto
	bujias	Texto

Cliente --> dni

Tabla reparaciones. Almacena los datos que indican el momento en que se realizó la reparación, datos técnicos y observaciones del taller y cliente. Un vehículo puede tener varias reparaciones pero en distinta fecha.



	Nombre del campo	Tipo de datos	
	matricula	Texto	Vehiculo -> matricula
	fecha	Fecha/Hora	
	km	Número	
	estado	Texto	
	ob_taller	Memo	
	ob_cliente	Memo	
	detalle	Texto	
	responsable	Texto	Empleado -> dni
	datos_tecnicos	Texto	
	numero	Número	
	ano	Número	

Tabla empleados. Recoge la información personal de los trabajadores del taller. Cada empleado es responsable de una reparación.


	Nombre del campo	Tipo de datos
	dni	Texto
	nombre	Texto
	apellido	Texto
	puesto	Texto
	antigüedad	Fecha/Hora
	clave	Texto
	telf	Texto

Tabla detalle_cargos. En ella reflejamos los trabajos realizados en la reparación indicada. La reparación puede tener varios conceptos de reparación. Son todos únicos de esa reparación.




	Nombre del campo	Tipo de datos	
	matricula	Texto	Reparaciones -> matricula, fecha
	fecha	Fecha/Hora	
	referencia	Número	
	tiempo_real	Moneda	
	descripcion	Texto	
	grupo	Número	
	subgrupo	Número	

Tabla detalle_repuestos. En ella almacenamos los repuestos utilizados en la reparación indicada. La reparación puede tener varios repuestos. Son todos únicos de esa reparación.





	Nombre del campo	Tipo de datos	
	matricula	Texto	Detalle Repuestos → matricula, fecha, referencia
	fecha	Fecha/Hora	
	referencia	Texto	
	proveedor	Texto	
	unidades	Moneda	
	precio_real	Moneda	
	descripcion	Texto	
	descuento	Moneda	
	precio_descuento	Moneda	

Tabla cargos_varios. Es la información genérica con los posibles conceptos de reparación que se pueden realizar. Indican el tiempo aconsejado por tarifa.


	Nombre del campo	Tipo de datos
	referencia	Autonumérico
	descripcion	Texto
	tiempo	Moneda

Tabla repuestos. Recoge la información de los repuestos, el proveedor y el precio de compra al suministrador.



	Nombre del campo	Tipo de datos	
	referencia	Texto	Proveedor → cif
	proveedor	Texto	
	descripcion	Texto	
	precio	Moneda	

Tabla proveedores. Almacena la información empresarial de los proveedores.




	Nombre del campo	Tipo de datos
	nombre	Texto
	cif	Texto
	direccion	Texto
	localidad	Texto
	provincia	Texto
	telf1	Texto
	telf2	Texto
	email	Texto
	fax	Texto
	cp	Número

Tabla facturas. Cuando una orden de reparación se cierra está preparada para facturarse. En la tabla almacenamos toda la información de esa factura.

	Nombre del campo	Tipo de datos	
	matricula	Texto	Reparaciones -> matricula, fecha
	fecha	Fecha/Hora	
	importe	Moneda	
	estado	Texto	
	iva	Número	
	precio	Moneda	
	numero	Número	
	ano	Número	
	fecha_factura	Fecha/Hora	

3.2.3.3 Diseño de la Interfaz de Usuario

Microsoft Visual Studio es una colección de herramientas que proporcionará al equipo de programación una sola fuente y un solo marco de trabajo para todo el desarrollo. Gracias **Visual Basic**, lenguaje de programación orientado a eventos, conseguiremos una Interfaz de Usuario atractiva y fácil de manejar.

Inicio de la Aplicación

Al arrancar la aplicación, el sistema pide identificación al usuario para acceder a la herramienta.



Ilustración 35. Inicio de la Aplicación


Vista Rápida del Cliente

La primera pantalla que se encuentra el usuario tras la conexión con la Base de Datos. En ella puede visualizar, mediante una búsqueda (DNI, Matrícula, Nombre, Apellidos), los datos personales de un cliente, los datos técnicos de sus vehículos y la información más importante respecto al mantenimiento de sus coches.

Archivo
Gestor BBDD
Ayuda



Cliente

DNI 52981185H
Nombre ALFREDO
Apellido 1 FERNANDEZ
Apellido 2 ARROYO
Dirección LOS PINOS 11
Localidad VILLALUENGA DE L
Provincia TOLEDO
CP 45520
Email ARMANDO -- 627050372
Teléfono 1 627825640
Teléfono 2 647410099





Vehículos

Matrícula 8610GPB
Marca PEUGEOT
Modelo 308 SPORT HDI
Tipo Motor D/9HZ
Año 05/08/2009
Ncc 4/1560
PF 11,45
Kw 80
Chasis VF34C9HZC9S131846

NO HAY DATOS


Selección Vehículo Cliente

Fecha Reservada 21/03/2016
Prox. Revisión 107500
Prox. ITV 20/09/2015
Prox. Distribución 180000

Vista Rápida


F. Combustible NO HAY DATOS
F. Aire C3282
F. Aceite HU716/2X
F. Habitaculo CU2940
Aceite Motor 10W40
Aceite Cambio 75W90
Bujías NO HAY DATOS



Neumáticos



205/55 R16 91V

Código Radio NO DATOS
Código Llave NO DATOS




Ref. Distribución NO HAY DATOS

180000 KM. O, 10 AÑOS (4/2019)

Historial Vehículo



13:11:27

BÚSQUEDA POR DNI: 52981185H

09/09/2015

Ilustración 36. Vista Rápida del Cliente

Luis Marino López Luque

81

Gestor de Base de Datos (Cliente, Vehículo)

En esta primera parte del Gestor de la Base de Datos, el usuario puede crear, modificar o eliminar registros de clientes y vehículos. El formulario presenta buscadores para encontrar el registro indicado a manipular.

Archivo **Cliente** **Vehículo** **Ayuda**

Cliente

DNI Nombre Apellido 1 Apellido 2

Dirección Localidad Provincia CP

Email Teléfono 1 Teléfono 2

Detalle Clientes

Búsqueda Clientes

☒ Todos los Clientes

☐ Por DNI

☐ Por Nombre

☐ Por Primer Apellido

Grabar Cliente

Grabar Vehículo

Nuevo Vehículo

Matrícula Tipo Motor

Marca Ncc

Modelo PF

Matriculación Kw

Chasis

Vista Rápida

Prox. Revisión Aceite Motor

Prox. Distribución F. Aire

Ref. Distribución F. Aceite

Neumáticos F. Habitaculo

Inf. Axone F. Combustible

Cod. Llave Bujías

Cod. Radio Aceite C/C

Intervalo Distribución:

Fecha Reservada ITV

Ilustración 37. Gestor de BD (Cliente y Vehículo)

Gestor de Base de Datos (Datos Corporativos, Proveedores, Empleados)

El formulario de Gestor de la Base de Datos permite también la actualización de los datos de empresa de TMB, de los proveedores y de los empleados del taller.

Archivo Ayuda

Tecnología Mecánica Benito

CIF Nombre N° Registro IVA % Precio/Hora €

Dirección Localidad Provincia Cp

Teléfono 1 Teléfono 2 Fax Email


Proveedores ELIGE UNA OPCIÓN

Empresa
(STD) STANDARD
ALFA (MEDIODIA
ALMACEN
ANTONIO J. VA
ARDASA 2000, S
AUTOMATIC

CIF Nombre Email

Dirección Localidad Provincia

Teléfono 1 Teléfono 2 Fax Cp



[← volver atrás](#)

Empleados ELIGE UNA OPCIÓN

DNI	Nombre	Apellido	Puesto	Teléfono	Antigüedad
53463131E	LUIS	LOPEZ	AYUDANTE	925531431	01/09/2000

DNI Puesto

Nombre Apellido

Antigüedad Teléfono

Proveedores

Empleados

Ilustración 38. Gestor de BD (TMB, Empleados, Proveedores)

Gestor de Almacén

La pantalla del almacén permite mantener los registros de los repuestos y los conceptos de reparación. Los registros pueden modificarse, eliminarse o crear uno nuevo.

Archivo Ayuda

Repuestos

ELIGE UNA OPCION

Ref	Descripción	Precio	Proveedor
5W40	ACEITE MOTOR	0	T/MANO NUEVO
▶ 5W40	ACEITE MOTOR	6,6	BRUDISA
5W50	ACEITE MOTOR	8,91	VEMARE
IRP135X55/5	ACEITE MOTOR	9,93	SEAT
CS SPEED	ACEITE SELECTOR	9	ALFA (MEDIODI
TEFLON	ADITIVO	24,81	VEMARE
TMB A/C.R.	ADITIVO C.REFRIGERACION	12	ALMACEN
500020	ADITIVO C/C	28,5	REAL
SE14194500B	AFORADOR	41,96	SEAT
7.02552.41.0	AFORADOR BOMBA	205	VEMARE
6N0919183B	AFORADOR DE DEPOSITO	128,96	SEAT
E1020M	AFORADOR/BOMBA	147,48	VEMARE
E10288M	AFORADOR/BOMBA	187,5	VEMARE
6K0880201B01C	AIRBAG DE VOLANTE	30	DESGUACES EL
GEMINI	ALARMA MONTADA FORD FIESTA	280	ALMACEN
0172436	ALETIN	18,05	OPEL
30850	ALFOMBRILLAS	0	REAL

Referencia

5W40

Borrar Repuesto

Descripción

ACEITE MOTOR

Modificar Repuesto

Proveedor

BRUDISA

Nuevo Repuesto

Precio

6.6


Opciones Repuestos

✓

✗

Repuestos

ELIGE UNA OPCION



Cargos Varios

ELIGE UNA OPCION

Descripción	Tiempo
REVISAR VIBRACION DE ESCAPE:	0
2ª ITV:	0
ACOPLAR RELE DE BOMBA	0
AJUSTAR BUJES DELANTEROS	0
AJUSTAR BUJES TRASEROS	0
AJUSTAR CAIDA DE RUEDAS	0
AJUSTAR CAPOT	0
AJUSTAR CORREA DE A/C.	0
AJUSTAR CORREA DE DIRECCION	0
AJUSTAR CORREA DE SERVICIO	0
AJUSTAR CORREAS	0
AJUSTAR FRENO DE MANO	0
AJUSTAR VARILLAJE DE PALANCA DE CAMBIOS	0
AJUSTAR/SUSTITUIR CORREA DE SERVICIO	0
AJUSTE DE BUJE TRASERO IZQ	0
▶ AJUSTE DE BUJES DELANTEROS	0
AJUSTE DE C.O.	0
AJUSTE DE CORREAS	0
AJUSTE DE RALENTI	0
AJUSTE Y REGULACION DEL CARBURADOR	0

Referencia

1.667

Borrar Cargo

Descripción

AJUSTE DE BUJES DELAN

Modificar Cargo

Tiempo

0

Nuevo Cargo

Opciones Cargos

✓

✗

Cargos Varios

ELIGE UNA OPCION




Ilustración 39. Gestor de Almacén

Luis Marino López Luque

84

Tras seleccionar un vehículo en la pantalla de **Vista Previa**, la aplicación accede a su libro de mantenimiento. En el formulario podemos visualizar el detalle de las reparaciones que se han realizado, ordenadas por fecha y con una descripción general orientativa. Observamos el detalle de los repuestos y de los conceptos de reparación, al igual que las observaciones del taller y del cliente.

Archivo Ayuda

Fecha Reservada 21/03/2016

Próxima ITV 20/09/2015

Proxima Distribución 180000

Próxima Revisión 107500

Vehículo

Matrícula 8610GPB

Tipo Motor D/9HZ

Marca PEUGEOT

Ncc 4/1560

Modelo 308 SPORT HDI

PF 11,45

Año 05/08/2009

Kw 80

Chasis VF34C9HZC9S131846

Detalle Reparaciones

Fecha	Km	Número	Año	Detalle
07/04/2015	92120	95	2015	NEUMATICOS
31/03/2015	92120	93	2015	RV/SF.1*

Fecha Reparación: 31/03/2015

Borrar Reparación

Modificar Reparación

Nueva Reparación

[← volver atrás](#)

Detalle Repuestos

PROVEEDOR

Descripción	Referencia	Uds	Precio
ACEITE MOTOR	10W40	5	5,75
FILTRO DE ACEITE	HU716/2X	1	13,83
FILTRO DE AIRE	C3282	1	18
GRAPA DE S.DE BARILLA SUP.	792834	1	0,25
LIQUIDO DE FRENOS	DOT4	1	0
KIT DE FRENOS DELANTEROS	8132300	1	137,58
TIRANTA DE SUSPENSION	50-08519	1	24,65

DATOS TÉCNICOS REPARACIÓN 10W40

Observaciones Cliente

REVISION,VIBRA EL VOLANTE ENTRE 100/110 KM./H.,Y TIRA UN POCO A LA IZQUIERDA.

Ilustración 40. Libro de Mantenimiento

Consola de Trabajo

La nueva orden de reparación va *montándose* cuando se añaden conceptos de reparación y repuestos. Las observaciones del cliente y del taller están presentes, ya que hay que tenerlas en cuenta a la hora de elegir los conceptos. En esta pantalla además tenemos acceso a los datos técnicos del vehículo por si hiciera falta su consulta.

Archivo Imprimir Plantillas Ayuda

Vehículo

Matrícula **8610GPB** DNI **52981185H** 31/03/2015

KM **92120** Detalle **RV/SF.1***

Prox. Rev. **107500** RSV **21/03/2016** ITV **20/09/2015**

Nº Hoja Trabajo **93** **2015** **HACIA RECAMBIOS**

Cargos Varios

Descripción
REPONER TAPA DE FILTRO DE HABITÁCULO
REVISAR FILTRO DE COMBUSTIBLE
SUSTITUIR ACEITE DE CAJA DE CAMBIOS Y FILTRO
SUSTITUIR CAJA FILTRO DE COMBUSTIBLE
SUSTITUIR FILTRO ACEITE C/CAMBIOS
SUSTITUIR FILTRO DE ACEITE DE CAJA DE CAMBIOS
SUSTITUIR FILTRO DE ACEITE
SUSTITUIR FILTRO DE AIRE
SUSTITUIR FILTRO DE COMBUSTIBLE
SUSTITUIR FILTRO DE GASOIL
SUSTITUIR FILTRO DE GASOLINA
SUSTITUIR FILTRO DE HABITACULO
SUSTITUIR FILTRO DE POLEN
SUSTITUIR FILTRO DESHIDRATADOR
SUSTITUIR PORTA FILTROS DE AIRE

---> **N** Tiempo **0** Group **1** **1**

Desc **SUSTITUIR FILTRO DE AIRE**

Detalle Trabajos Realizados

Descripción	Tmp	G	SG
RV TMB	0	1	1
PREPARACIÓN DEL TRABAJO	0	1	2
AUTODIAGNOSIS DE SISTEMAS Y RESET	0	1	3
SUSTITUIR GRAPA VARILLA DE CAPOT	0	1	4
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE	0	1	5
SUSTITUIR FILTRO DE AIRE	0	1	6
SUSTITUIR FILTRO DE HABITACULO	2,1	1	7
REPARACION DE FRENOS:	0	4	1
SUSTITUIR KIT DE FRENOS DELANTEROS	0	4	2
SUSTITUIR LIQUIDO DE FRENOS	0	4	3
SUSTITUIR TIRANTA DE SUSPENSION D/D.	1,8	5	1

SUSTITUIR FILTRO DE AIRE 1 6 **Borrar**

DATOS TÉCNICOS REPARACIÓN 10W40

Observaciones Cliente

REVISION,VIBRA EL VOLANTE ENTRE 100/110 KM./H.,Y TIRA UN POCO A LA IZQUIERDA.

Grabar Reparación

volver atrás

Ilustración 41. Consola de Trabajo

Facturación

Una hoja de reparación se factura cuando se encuentra cerrada, es decir los trabajos ya se han realizado. En este último paso se ofrece la posibilidad de modificar todos los conceptos, los precios, tiempos y unidades. Todos estos cambios se actualizarán en el Libro de Mantenimiento cuando se termine la factura.

Archivo Ayuda

Vehículo

Matrícula **8610GPB** DNI **52981185H** RSV **21/03/2016** ITV **20/09/2015**

KM **92120** Prox. Revisión **107500** Detalle **RV/SF.1*** Fecha **09/09/2015**

☒ REVISION TMB

☒ PREPARACION DEL TRABAJO

☒ AUTODIAGNOSIS Y RESET

Detalle Repuestos

Ref	Descripción	Uds	Precio	Dto	Importe
10W40	ACEITE MOTOR	5	5,75	0	5,75
HU716/2X	FILTRO DE ACEITE	1	13,83	0	13,83
C3282	FILTRO DE AIRE	1	18	0	18
792834	GRAPA DE S.DE BARILLA SU	1	0,25	0	0,25
DOT4	LIQUIDO DE FRENOS	1	0	0	0
8132300	KIT DE FRENOS DELANTEROS	1	137,58	0	137,58
50-08519	TIRANTA DE SUSPENSION	1	24,65	0	24,65

TOTAL MATERIALES 223,06 Dto Normal

Datos Factura

Precio/Hora € **37** IVA % **21** Total IVA € **77,15**

TOTAL REPARACION € 367,36 TOTAL FACTURA € 444,51

Detalle Trabajos Realizados

Descripción	Tmp	G	SG
RV TMB	0	1	1
PREPARACIÓN DEL TRABAJO	0	1	2
AUTODIAGNOSIS DE SISTEMAS Y RESET	0	1	3
SUSTITUIR GRAPA VARILLA DE CAPOT	0	1	4
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE	0	1	5
SUSTITUIR FILTRO DE AIRE	0	1	6
SUSTITUIR FILTRO DE HABITACULO	2,1	1	7
REPARACION DE FRENOS:	0	4	1
SUSTITUIR KIT DE FRENOS DELANTEROS	0	4	2
SUSTITUIR LIQUIDO DE FRENOS	0	4	3
SUSTITUIR TIRANTA DE SUSPENSION D/D.	1,8	5	1

TIEMPOS 3,9 **TOTAL MANO DE OBRA** 144,3

Observaciones Taller

RUIDO POR ZONA DE DISTRIBUCION.
M,S/I:P1461

[volver atrás](#) Número de Factura 123 2015

Ilustración 42. Facturación

3.2.4 Implementación del Sistema de Información

Con el diseño de la Base de Datos y el Interfaz de Usuario ya podemos construir el sistema de información. En este capítulo también haremos referencia a la implantación de la Aplicación de Escritorio en el entorno de operación.

3.2.4.1 Construcción del Sistema de Información

Conocer tanto el entorno de desarrollo de la aplicación como su entorno de operación, son puntos necesarios para la construcción del Software de Escritorio.

3.2.4.1.1 Entorno de Desarrollo

Microsoft Visual Studio es un entorno para sistemas operativos Windows cuya finalidad es la creación de aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte .NET. El objetivo de esta colección de herramientas es proporcionar al equipo de programación una sola fuente y un solo marco de trabajo para todo el desarrollo.

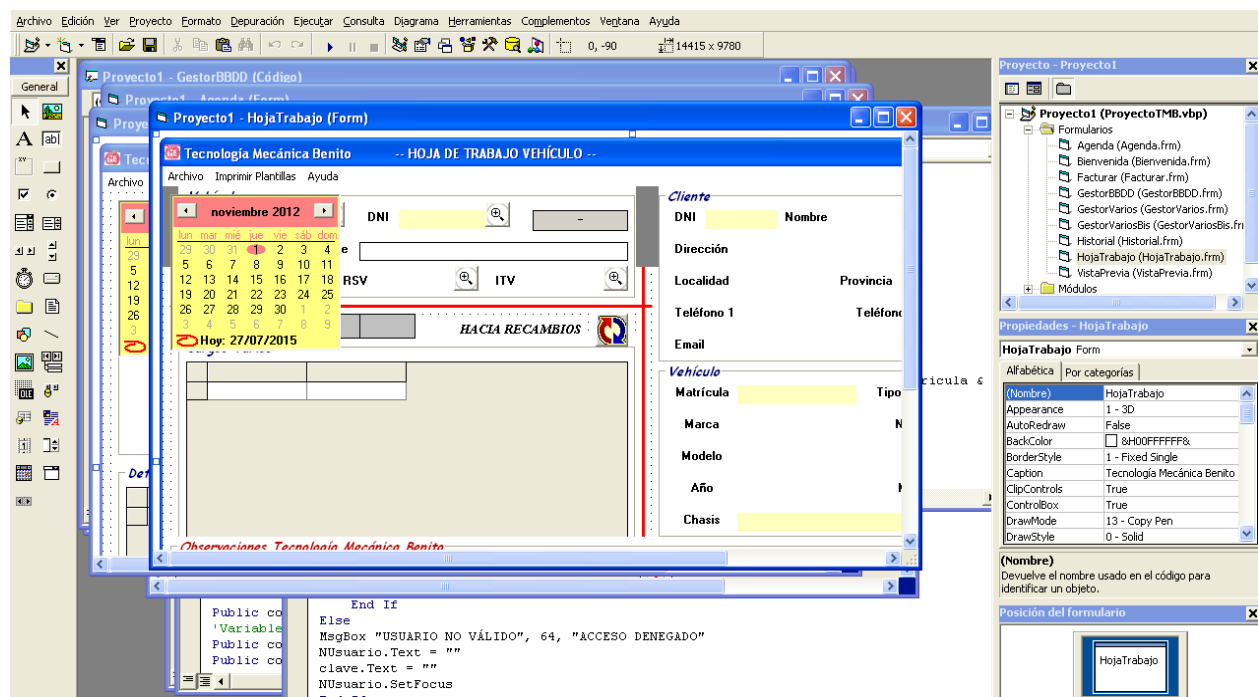


Ilustración 43 IDE Visual Studio

Como explicamos anteriormente, **Visual Basic** será el lenguaje de programación elegido, funcionará *encima* de **Access** y creará un interfaz de usuario sencilla y muy intuitiva al ser un lenguaje dirigido por eventos.

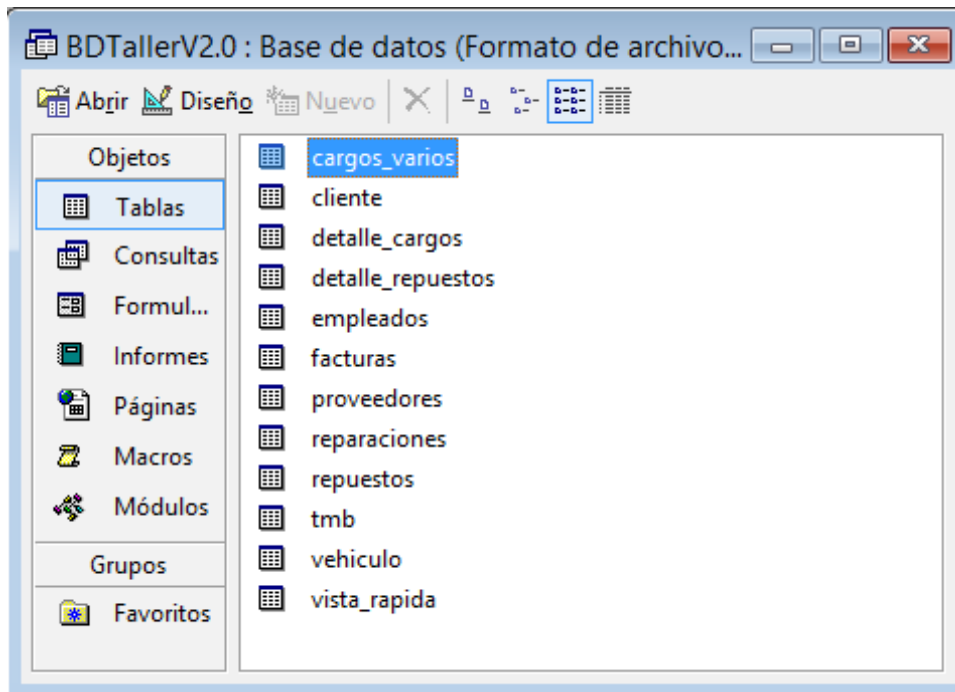


Ilustración 44. Access - Gestor BBDD

La empresa poseía una Suite Ofimática (**Microsoft Office - Access**) con la que creamos nuestra Base de Datos. **Visual Basic** manipula directamente los registros de la Base de Datos a través de los *Recordset* para posicionarse.

3.2.4.1.2 Entorno de Ejecución

Lo más destacable de la relación que hemos creado entre Access y Visual Basic es el Acceso a los Datos. **OLE DB** es una interfaz de programación de bajo nivel diseñada para acceder a diversos orígenes de datos. Puede soportar tanto orígenes de datos relacionales como no relacionales.

Pero no utilizamos OLE DB directamente, nos servimos de **ADO** para tratar con la capa de OLE DB. Cuando hablamos de ADO nos referimos a un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Los objetos **Connection**, **Recordset** y **Field** serán los más utilizados en esta aplicación.

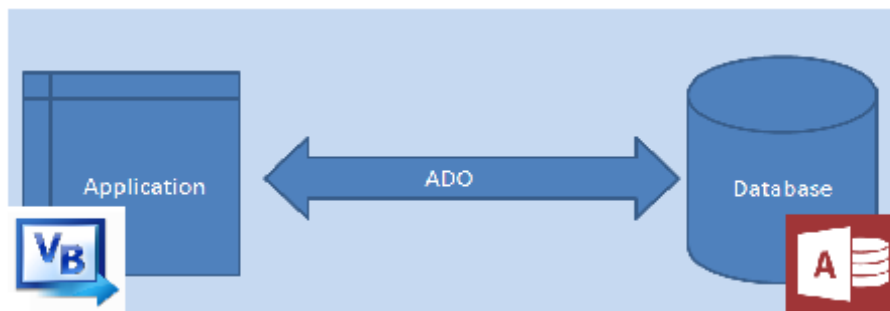


Ilustración 45. Acceso a Datos

La Base de Datos y la Aplicación de Escritorio estarán almacenadas en el mismo equipo, por lo que será importante conocer la dirección física de la Base de Datos para establecer la conexión.

3.2.4.1.3 Codificación de los Niveles de la Aplicación

Diseñada la Interfaz de Usuario vamos analizar los aspectos de codificación más importantes de la Aplicación de Escritorio.

Pantalla de Identificación

Al ejecutar el programa, la aplicación solicita la autenticación del usuario. No hay conexión con la Base de Datos hasta que no esté identificado el administrativo.

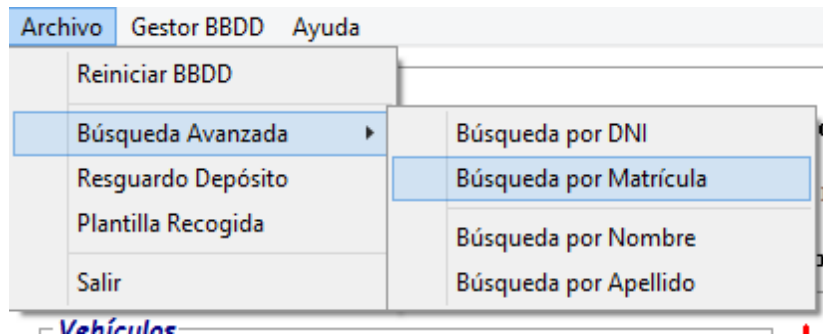
En este caso, la Base de Datos tiene contraseña, y el programador necesita conocerla en desarrollo para implementar la conexión.

```
strBase = "C:\BDTallerV2.0.mdb"
strClave = Bienvenida.clave.Text
conexion = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source = " & strBase & ";" & " Jet OLEDB:Database "
Password = " & strClave"
```

Ilustración 46. Conexión con la Base de Datos

Procesos de Búsqueda

Para facilitar la labor del usuario existen menús con varios criterios de búsqueda. El usuario elige la opción y una caja de texto requerirá el término a buscar.



La aplicación recogerá el valor y aplicará la *select* para rellenar el *recordset* (ADO). Después el programa visualizará los datos en cajas de textos para los datos sencillos, y en *Grid* para los datos que tengan varios campos por registro.

```
Private Sub MenuMatricula_Click()  
'OBTENEMOS LA MATRICULA ATRAVES DEL INPUTBOX  
consultaMatricula = InputBox("INTRODUZCA MATRÍCULA A BUSCAR:", "BÚSQUEDA MATRÍCULA")  
'CERRAMOS RECORDSET PARA RELLENAR VISTAPREVIA
```

Ilustración 47. Procesos de Búsqueda

Visualización de los datos

Definimos *Grid* como las *tablas* donde descargar los datos que contienen los *recordset*. Se utilizan porque son intuitivos, y cada campo de cada fila son seleccionables para trabajar con ellos.

Detalle Trabajos Realizados	
Descripción	Tiempo
RV TMB	0
PREPARACIÓN DEL TRABAJO	0
▶ AUTODIAGNOSIS DE SISTEMAS Y RESET	0
SUSTITUIR GRAPA VARILLA DE CAPOT	0
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE	0
SUSTITUIR FILTRO DE AIRE	0
SUSTITUIR FILTRO DE HABITACULO	2,1
REPARACION DE FRENOS:	0
SUSTITUIR KIT DE FRENOS DELANTEROS	0
SUSTITUIR LIQUIDO DE FRENOS	0
SUSTITUIR TIRANTA DE SUSPENSION D/D.	1,8

Ilustración 48. Tabla vista Usuario

```

mirecordset_detalle_cargos.Open "select  descripcion, referencia, tiempo_real,

If mirecordset_detalle_cargos.BOF Or mirecordset_detalle_cargos.EOF Then
    'MsgBox "NO HAY CARGOS VARIOS", 64, "ATENCIÓN"
    mirecordset_detalle_cargos.Close
Else
    Set Historial.GridTrabajos.DataSource = mirecordset_detalle_cargos
    Historial.GridTrabajos.Columns(0).Width = 4500
    Historial.GridTrabajos.Columns(1).Width = 1500
    Historial.GridTrabajos.Columns(2).Width = 800
    Historial.GridTrabajos.Columns(0).Caption = "Descripción"
    Historial.GridTrabajos.Columns(1).Caption = "Referencia"
    Historial.GridTrabajos.Columns(2).Caption = "Tiempo"
    Historial.GridTrabajos.Columns(1).Visible = False
    Historial.GridTrabajos.Columns(3).Visible = False
    Historial.GridTrabajos.Columns(4).Visible = False
End If

```

Ilustración 49. Tabla Vista Diseño

Actualizar la Base de Datos

Para formalizar cambios en nuestra Base de Datos, necesitamos un *recordset* que recoja los datos a almacenar, ya sean de cajas de texto o *Grid*. La idea es ir *moviendo* los registros del *recordset*, y almacenar en cada campo (*Fields*) de cada registro el dato correspondiente.

```

If nombre.Text = "" Then
    nombre.Text = "NO HAY DATOS"
End If
mirecordset_cliente.Fields("nombre") = UCase(nombre.Text)

```

Ilustración 50. Recogida de Datos

Facturación

Con sencillas fórmulas matemáticas y datos predefinidos conseguimos que el programa calcule el importe total de la factura. Hay datos que conocemos de antemano, como el precio de la hora y el IVA (Campos modificables en el Gestor de BD), con los que calcular el resto de valores. También aprovechamos el lenguaje de programación para dar formato a los resultados.

Datos Factura			
Precio/Hora €	37	IVA %	21
		Total IVA €	77,15
TOTAL REPARACION €		367,36	TOTAL FACTURA €
		444,51	




Ilustración 51. Facturación vista Usuario

```

-----
VTotalFactura = VTotalRepuestos + VTotalTrabajos
TotalReparacion.Text = Format(VTotalFactura, "###,###,##0.00")
TotalIva = VTotalFactura * Viva
VTotalFactura = VTotalFactura + TotalIva
Tiva.Text = Format(TotalIva, "###,###,##0.00")
TotalFactura.Text = Format(VTotalFactura, "###,###,##0.00")

```

Ilustración 52. Facturación vista Diseño

3.2.4.2 Implantación del Sistema de Información

El entorno de desarrollo es similar al entorno de operación, **Visual Studio** es un entorno para sistemas Windows y no habrá problemas para implantar la Aplicación. Además **Access** pertenece a **Microsoft Office** por lo que la sinergia con el software será máximo. Solo se instalará en un equipo, por lo que satisfacemos los requisitos de Software y Hardware mínimos.

Aunque se instalan todos los recursos en un mismo equipo, el código del programa debe conocer la ubicación de la Base de Datos y de las imágenes a visualizar.

3.2.4.2.1 Organización de Contenidos

El paquete de instalación de Visual Basic preguntará donde instalar la aplicación, sin restricción alguna. Pero el programador debe saber de antemano la dirección de la Base de Datos y su contraseña, si la tuviera, para conectar con ella. El programa también realiza llamadas a imágenes para mostrarlas, así que también es necesario conocer donde se encuentran.

► Este equipo ► Windows (C:) ►





Nombre	Fecha de modifica...	Tipo	Tamaño
 BDTallerV2.0	09/09/2015 13:21	Archivo MDB	6.916 KB
 Imagenes	10/09/2015 12:32	Carpeta de archivos	
 Windows	09/09/2015 16:40	Carpeta de archivos	
 Archivos de programa (x86)	08/09/2015 12:45	Carpeta de archivos	

Ilustración 53. Organización de Contenidos

Lo ideal es que todas las imágenes se encuentren en una carpeta y a la misma altura que la Base de Datos. Para no complicar la fase de pruebas se suelen situar directamente en **la raíz del sistema**.

3.2.4.2.2 El Entorno de Operación

Instalación en el entorno

El programador después de testar la aplicación en su entorno de desarrollo, ejecuta asistente de empaquetado y distribución de Visual Studio. El primer paso de la creación del **Paquete de Instalación** es compilar la Aplicación para crear un **Ejecutable**.

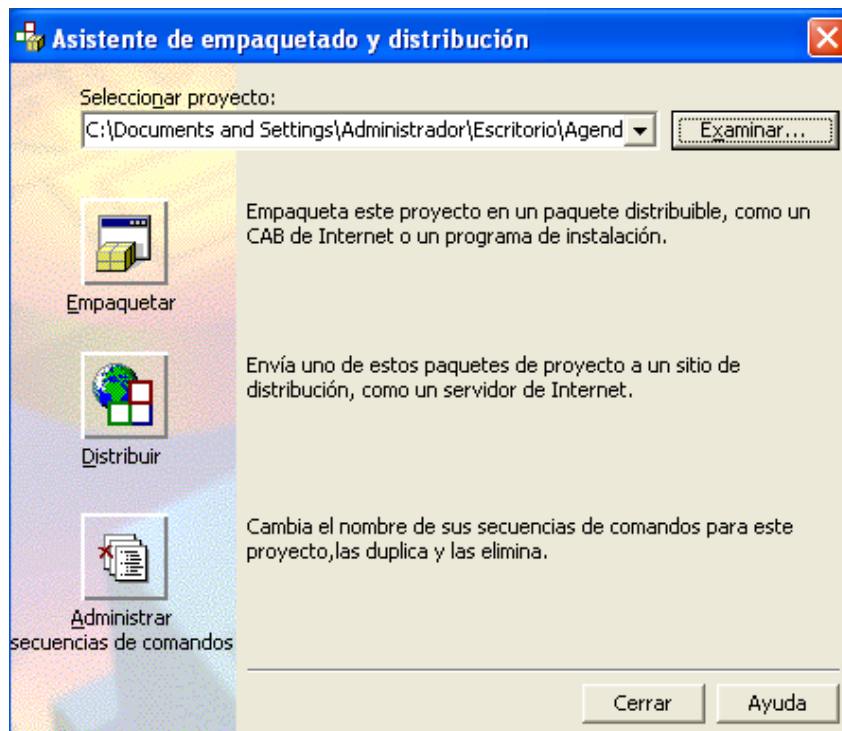


Ilustración 54. Asistente de empaquetado y distribución

Tras crear el ejecutable el asistente solicita la carpeta donde ubicar el paquete de instalación y los archivos a incluir en la instalación final del programa.



Ilustración 55. Carpeta del paquete

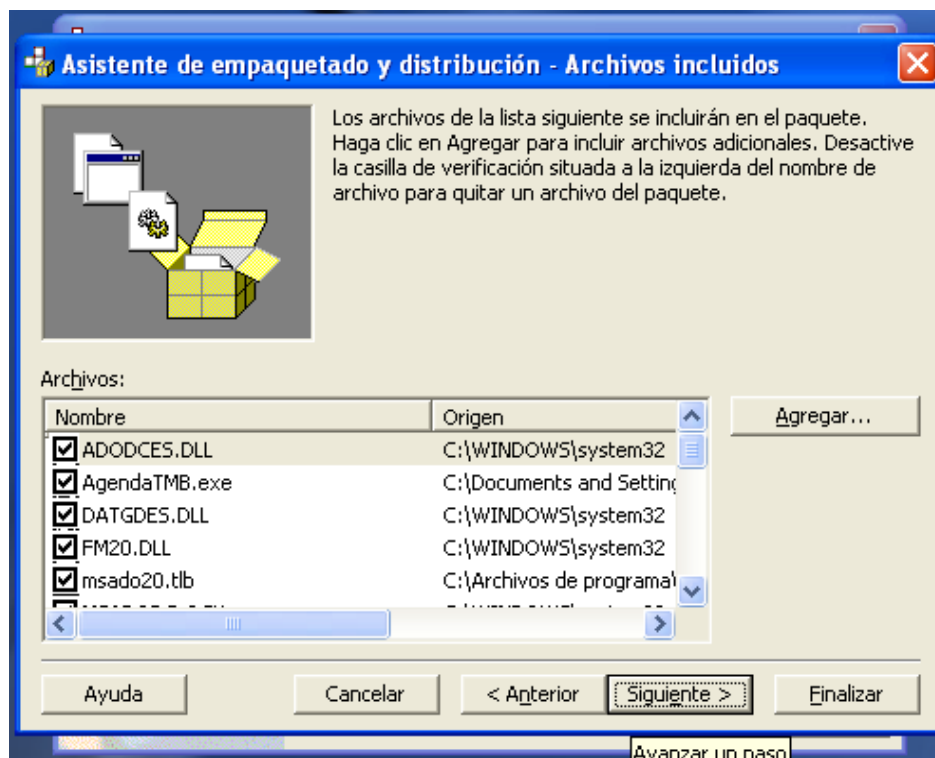


Ilustración 56. Archivos incluidos

Con el paquete de distribución creado, sólo falta ejecutarlo en el entorno de operación y seguir los pasos del instalador, elegir la dirección de instalación, el nombre de la aplicación, etc. Si la instalación es correcta tenemos nuestro programa listo para ejecutarse.

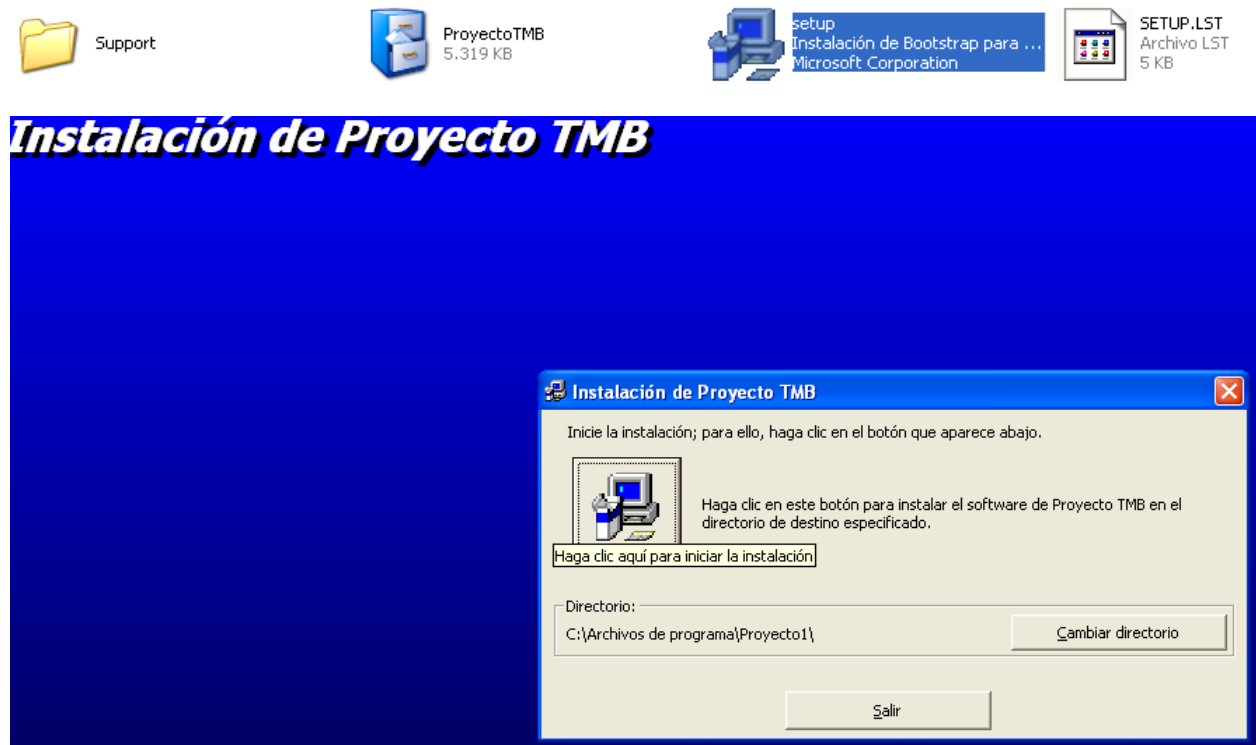


Ilustración 57. Instalación del Proyecto

Carga y Migración de los Datos

En primera instancia debe estar creada la Base de Datos en blanco, y a medida que se trabaja con la aplicación crecer en contenidos. La carpeta con las imágenes también debe estar creada con las imágenes a visualizar.

No existe **Carga de Datos** si no es por un problema que necesita reinstalación de la Base de Datos, y la **Migración** sólo se efectúa en los respaldos realizados bajo indicación del Documento de Seguridad.

3.3 Libro de Mantenimiento online

Tras estudiar la solicitud del Taller realizaremos el Análisis, Diseño e Implementación del sistema de información, Portal Web del Taller, siguiendo la metodología MÉTRICA Versión 3.

3.3.1 La Solicitud

TMB busca con el **Libro de Mantenimiento online** dos objetivos principales: Ampliar su cartera de clientes y consolidar su relación con los que ya tiene. Partiendo de una Base de Datos procedente de su aplicación de escritorio, conseguiremos que el cliente consulte el historial de reparaciones de su vehículo a través de un Portal Web.

TMB distingue **dos ámbitos** para el sitio Web, el externo o **público**, zona del portal accesible por todos los usuarios, y el interno o **privado**, una pequeña zona privada para los clientes del taller.

Para TMB también es importante crear un interfaz cómodo, sencillo y eficiente para interactuar con el cliente, y conseguir un sistema seguro y robusto frente a fallos e intrusos. Todo ello evitando en la medida de lo posible un alto consumo de recursos software y hardware.

3.3.2 Análisis del Sistema de Información

En este apartado analizaremos la situación actual del sistema, desarrollaremos el catálogo de requisitos y visualizaremos los Diagramas de Casos de Uso para aquellos que sean funcionales.

3.3.2.1 Definición del Sistema

El desarrollo de la aplicación no afectará a la estructura organizativa de la empresa, ya que actualmente trabaja con el software de escritorio y el nuevo sistema solo utilizará los respaldos de la Base de Datos que le proporcionará el Taller, por tanto, el desarrollo del nuevo software no impedirá que los empleados de la empresa sigan desempeñando sus funciones con normalidad.

Será necesario adecuar la Base de Datos proporcionada por TMB para mostrar la información al cliente a través de la Web. Recibiremos una Base de Datos completa en **Access**, con información relativa a todo el funcionamiento de la empresa, clientes, proveedores, almacén, reparaciones,

vehículos... y extraeremos sólo la información relativa al mantenimiento de los vehículos (**Anexo2**).

El portal Web distinguirá dos zonas: **Pública** y **Privada**

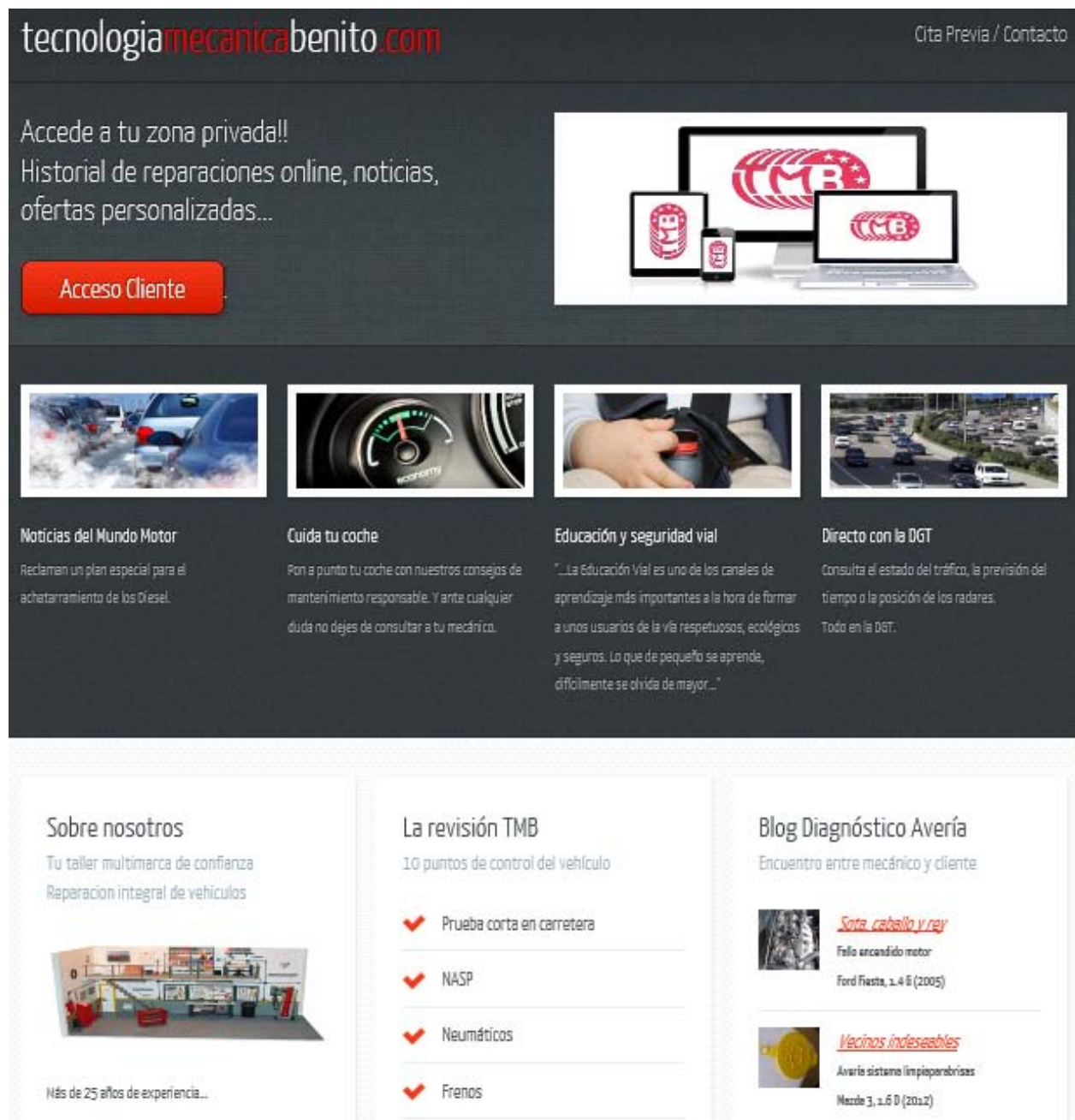


Ilustración **Ámbito Público**

En el ámbito público se permitirá el libre acceso, donde los usuarios podrán acceder a la información del Taller, contactar con la empresa, consultar noticias y consejos del mundo del

motor, consultar el tráfico y leer las entradas del blog de encuentro **Diagnostico Avería** para conocer el día a día del Taller.



Control Acceso

Nombre De Usuario

Contraseña

Entrar

TMB le ha proporcionado una clave secreta para acceder a su zona privada.
Por razones de seguridad, si ha olvidado su clave, o la ha extraviado, póngase en contacto con el taller.

Ilustración Acceso Zona Privada

Para acceder a la zona privada, existirá una pasarela de identificación donde se solicitarán los datos de acceso a los clientes registrados. Una vez realizado el proceso de autenticación, el cliente podrá consultar el historial de mantenimiento de sus vehículos y beneficiarse de las ofertas personalizadas.



Ilustración Ámbito Privado

Tras considerar el alcance del sistema y los objetivos del estudio de solicitud debemos analizar la situación actual, ya que nos ayudará a concretar que podemos aprovechar del antiguo sistema para el futuro proyecto. No olvidemos que el nuevo sistema recibirá información relativa a las reparaciones de los vehículos del sistema antiguo, y que ambos sistemas van a funcionar a la vez.

Objetivos del estudio de la situación actual:

Decidir el modo en que los distintos usuarios van a acceder al sistema.

Determinar la arquitectura general que se va a implantar.

Tener una información completa sobre las necesidades de los usuarios para comprender qué problemas plantea el desarrollo de la aplicación.

3.3.2.2 La Situación Actual

La empresa dispone de una aplicación de escritorio mediante la cual registra y gestiona toda la información necesaria para la gestión de un taller de automoción. Almacena tanto datos económicos (facturas, almacén, proveedores) como personales (clientes, empleados), sin olvidar los datos técnicos referentes a las reparaciones de los vehículos.

Actualmente, TMB tiene la siguiente infraestructura: A la aplicación solo accede el personal de oficina, trabajando en todo momento offline.



No hay arquitectura de red, por lo que no existe flujo de datos entre los clientes y TMB.

El principal desafío será el diseño de una nueva arquitectura que permita el acceso a los clientes a sus datos de reparaciones, sin perder la capacidad de trabajo de la aplicación de escritorio.

3.3.2.3 Definición de los Requisitos del Sistema

Observando a los usuarios en su entorno de trabajo y realizado entrevistas, conseguimos que los clientes descubran, revelen, articulen y comprendan los requisitos que desean.

Descripción de las necesidades básicas

Facilidad de manejo y gestión
Disponibilidad del sistema
Seguridad en la operaciones y la transmisión de datos
Gestión del ámbito privado
Gestión del ámbito público
Recursos software y hardware mínimos

Facilidad de manejo y gestión

- La interfaz debe ser manejable y accesible a cualquier tipo de cliente sin importar sus conocimientos de informática.
- Facilidad de gestión por parte de la empresa tanto de la parte pública como de la privada.

Disponibilidad del sistema

- El sistema desarrollado debe estar disponible siempre, no solo en horario laboral. El servidor debe dar servicio a los clientes en cualquier momento.

Seguridad en las operaciones y en la transmisión de datos

- Los usuarios del sistema deben tener los permisos de acceso de TMB.
- El envío de la BD al servidor debe ser encriptado, evitando la posible pérdida de datos y asegurando la confidencialidad de los mismos.
- Conseguir un riesgo mínimo ante ataques de usuarios ajenos al sistema.

La empresa dispone de un Documento de Seguridad (**Anexo 5**) en el que responde a los puntos siguientes:

- Continuidad: recuperar los datos
- Política de seguridad y control
- Integridad de los datos

3.3.2.4 Catalogación de Requisitos

Basándonos en las operaciones que debe realizar el sistema podemos catalogar los requisitos de la aplicación de la siguiente forma:

Requisitos Funcionales

Descripción	Prioridad
El Cliente registrado debe tener un acceso a la aplicación mediante su identificación, única e intransferible, (login y password)	↑↑
La aplicación permitirá un acceso directo al formulario de cita previa / contacto	↑↑
La aplicación sugerirá noticias del mundo motor a consultar	⇒
Se debe permitir la actualización de las noticias del mundo motor por parte del personal de TMB	⇒
La aplicación sugerirá consejos de mantenimiento del vehículo a consultar	⇒
Se debe permitir la actualización de los consejos de mantenimiento por parte del personal de TMB	⇒
La aplicación sugerirá consejos de educación vial a consultar	⇒
Se debe permitir la actualización de los consejos de educación vial por parte del personal de TMB	⇒
La aplicación permitirá un acceso directo a la situación del tráfico	⇒
La aplicación permitirá el acceso a la información comercial de TMB	↑
La aplicación permitirá el acceso a la información legal del sitio	↑
La aplicación sugerirá entradas del blog de apoyo a consultar	⇒
Se debe permitir la actualización de los enlaces de entradas al blog por parte del personal de TMB.	⇒
El cliente registrado debe acceder a cada reparación de cada uno de sus vehículos	↑↑
El cliente registrado debe acceder a sus ofertas personalizadas	↑↑
El cliente registrado debe acceder al formulario de contacto de la parte privada	↑↑
El cliente registrado debe acceder a los avisos legales del sitio y protección de datos	↑
El cliente registrado debe tener la posibilidad de cerrar sus sesión	↑↑
Se debe permitir la actualización de las ofertas personalizadas por parte del personal de TMB.	↑↑

Requisitos de Implantación

RECURSOS SOFTWARE Y HARDWARE MINIMOS

Puesto que se quiere que la aplicación sea de uso general y que no implique ni amplios conocimientos de informática ni grandes equipos para hacerla funcionar, se ha diseñado de forma que pueda *correr* en todo tipo de dispositivos. (**HTML5, PHP, MySQL**)

DEFINICION DEL ENTORNO TECNOLÓGICO DEL SISTEMA

El sistema estará compuesto por un servidor global, proporcionado por un proveedor de alojamiento Web, en donde se almacenarán tanto los datos a consultar como los archivos necesarios de la arquitectura del Portal. Los clientes accederán a los servicios desde sus dispositivos.

En cuanto a la gestión de los datos el sistema utilizará MySQL, mostrando los resultados en la zona privada mediante la funcionalidad de HTML5. La Base de Datos en esta versión de la aplicación será de solo lectura, por lo que el riesgo de inconsistencia en los datos es mínimo.

Requisitos de Disponibilidad

La aplicación debe permitir el acceso concurrente a todos los usuarios que tengan acceso.

Requisitos de Rendimiento

La elección de un buen alojamiento marcará el rendimiento de la aplicación, sin olvidar la importancia de un buen diseño Web para que los distintos navegadores muestren la información de forma eficaz.

Requisitos de Seguridad

La empresa dispone de documento de seguridad, por lo que los puntos relativos a copias de seguridad, respaldos y errores están cubiertos. El proveedor de alojamiento se encargará de la protección de los archivos en su servidor.

La transmisión de la BD de la empresa al servidor debe ser segura y con las contraseñas encriptadas. El acceso a la zona privada debe tener un alto grado de seguridad frente a usuarios no registrados.

3.3.2.5 Casos de Uso

En el caso de orientación a objetos se especifican, además, los casos de uso asociados a los requisitos funcionales.

Los casos de uso son una técnica de especificación de requisitos válida tanto en desarrollos estructurados como en orientación a objetos, aunque en este último caso se propone como técnica obligatoria al ser necesaria como referencia a lo largo de todo el ciclo de vida.

Vamos a identificar los actores, casos de uso y realizaremos una breve explicación de cada caso.

En el portal Web que se va a desarrollar aparecerán tres tipos de Usuarios / Actores:

- **Usuario General.** Aunque no está registrado puede acceder a la parte pública sin restricciones y ponerse en contacto con la empresa.
- **Usuario Registrado.** A parte de disfrutar de la zona pública (en cuyo caso será considerado como Usuario General), tendrá acceso a su Área Privada para consultar los datos de las reparaciones de sus vehículos.
- **Administrador del Sistema.** Todo el trabajo de actualización de la Base de Datos se realiza en la aplicación de escritorio. Las partes del Portal actualizables son de carácter público fuera de la Base de Datos, pertenecientes a los archivos de arquitectura Web dentro del servidor global. Lo consideraremos un actor especial que interactúa en el sistema pero dentro del servidor privado.

A continuación, se muestran los **Diagramas de Casos de Uso** que representan las principales funcionalidades que el sistema deberá tener de cara a los distintos usuarios.

Descripción Gráfica

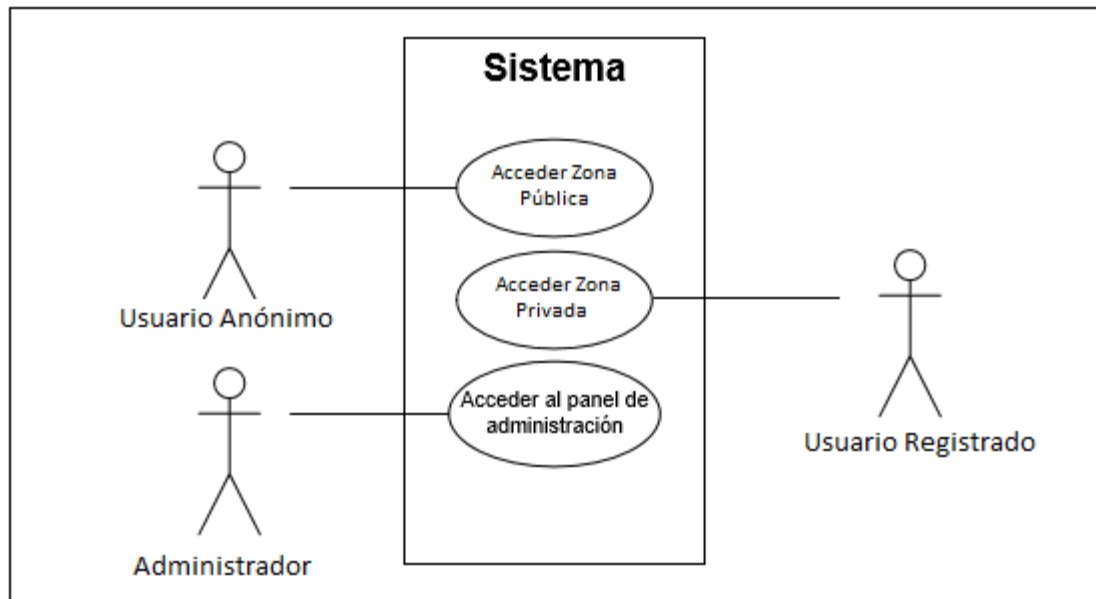


Ilustración 58 Nivel 0 - Diagrama de Casos de Uso

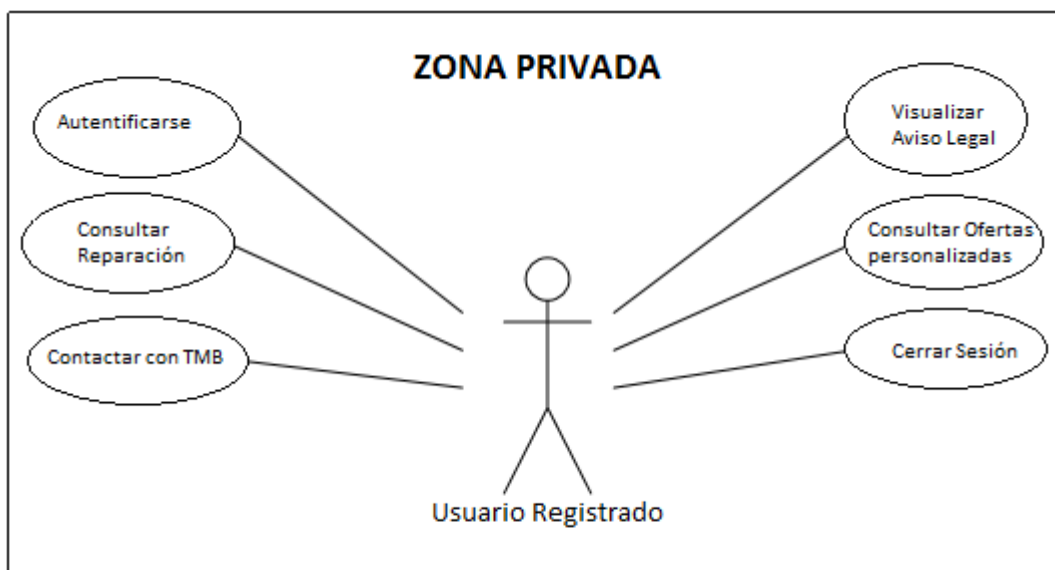


Ilustración 59 Nivel 1 - Diagrama de Casos de Uso "Acceder Zona Privada"

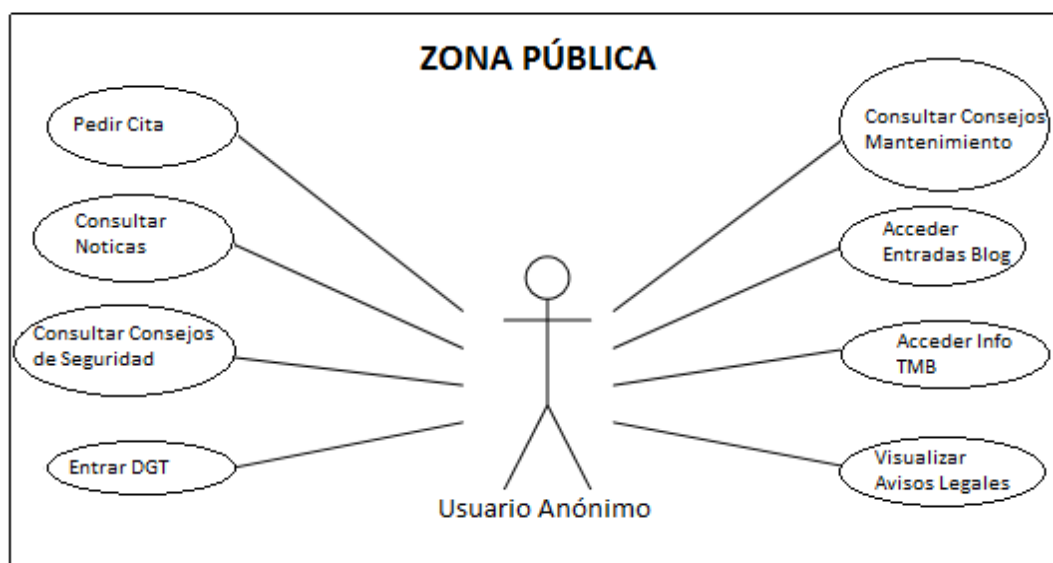


Ilustración 60 Nivel 1 - Diagramas de Caso de Uso "Acceder Zona Pública"

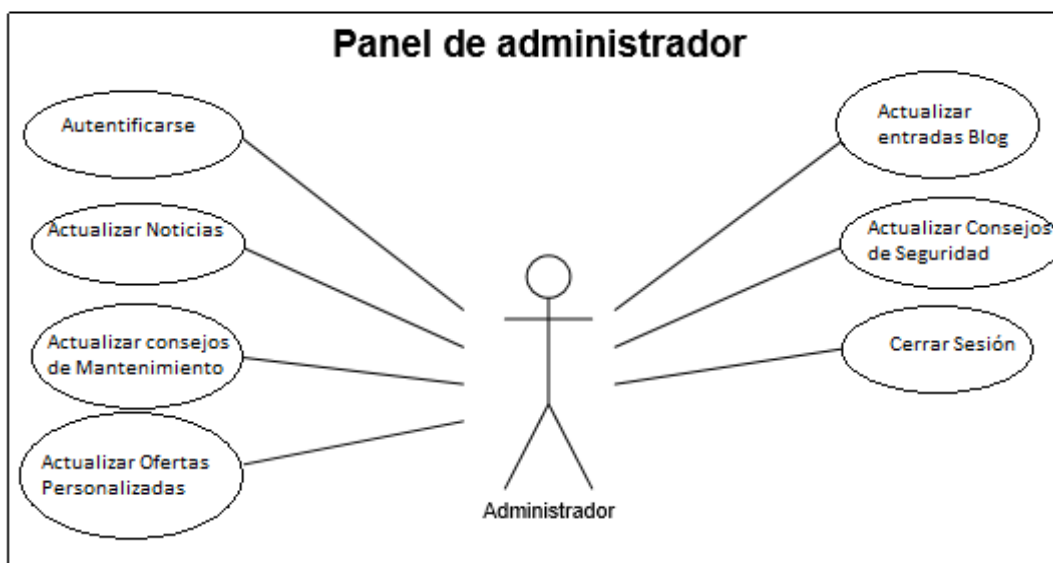


Ilustración 61 Nivel1 - Diagrama de Casos de Usos "Acceder al Panel de Administrador"

Descripción Textual

Caso de Uso	Autentificarse
Actor	Usuario Registrado
Objetivo	Conseguir acceso a su zona privada
Descripción	Dentro de la pasarela de autenticación, el usuario introducirá las claves proporcionas por TMB, login y password, para poder acceder a su zona privada.

Caso de Uso	Consultar Reparación
Actor	Usuario Registrado
Objetivo	Visualizar el detalle de una reparación
Descripción	Dentro del ámbito privado, el usuario podrá visualizar el detalle de cualquier reparación de sus vehículos. A través de sencillas tablas de selección, elegirá un vehículo de su propiedad y posteriormente la reparación a consultar.

Caso de Uso	Contactar con TMB
Actor	Usuario Registrado
Objetivo	Ponerse en contacto con la empresa
Descripción	El usuario registrado puede ponerse en contacto con el Taller mediante formulario, teléfono, mail. En su zona privada posee un enlace directo para conseguir está comunicación. También el enlace muestra información geográfica de la dirección del Taller (Mapa Interactivo).

Caso de Uso	Visualizar Aviso Legal
Actor	Usuario Registrado
Objetivo	Conocer los derechos y deberes del servicio TMB
Descripción	El usuario debe conocer las líneas legales principales de la utilización del servicio TMB. A través de un enlace conocerá sus derechos y deberes, así como la Ley de Protección de Datos en caso de reclamación.

Caso de Uso	Consultar Ofertas Personalizadas
Actor	Usuario Registrado
Objetivo	Disfrutar de las ventajas del servicio TMB
Descripción	El Usuario del Servicio Web TMB tiene acceso a ofertas y noticias personalizadas a través de un menú deslizante en la pantalla principal de su zona privada.

Caso de Uso	Cerrar Sesión
Actor	Usuario Registrado
Objetivo	Abandonar de forma segura su zona personal.
Descripción	Como medida de seguridad, el ámbito privado del portal ofrece al usuario registrado un enlace para abandonar de forma segura su sesión.

Caso de Uso	Pedir Cita
Actor	Usuario Anónimo
Objetivo	Contactar con el Taller
Descripción	Cualquier usuario del ámbito público de la Web puede ponerse en contacto con el Taller mediante formulario, teléfono, mail. También el enlace muestra información geográfica de la dirección del Taller.

Caso de Uso	Consultar Noticias
Actor	Usuario Anónimo
Objetivo	Conocer las últimas noticias del Mundo Motor
Descripción	El usuario encontrará enlaces a las últimas noticias del mundo del automóvil, tanto enlaces directos a las fuentes de la noticias como enlaces del blog de encuentro de TMB.

Caso de Uso	Consultar Consejos de Seguridad
Actor	Usuario Anónimo
Objetivo	Informar a los usuarios de la importancia de la seguridad al volante.
Descripción	El usuario encontrará enlaces con noticias y consejos de seguridad relacionados con el mundo del automóvil.

Caso de Uso	Entrar DGT
Actor	Usuario Anónimo
Objetivo	Conocer el estado del tráfico
Descripción	El usuario tiene la posibilidad de conocer tanto el tráfico en tiempo real como las previsiones de lluvia a través de un enlace directo de la DGT.

Caso de Uso	Consultar Consejos de Mantenimiento
Actor	Usuario Anónimo
Objetivo	Informar a los usuarios de la importancia del buen mantenimiento de su vehículo.
Descripción	El usuario encontrará enlaces a consejos de mantenimiento del mundo del automóvil, tanto enlaces directos a las fuentes de la noticias como enlaces del blog de encuentro de TMB.

Caso de Uso	Acceder Entradas Blog
Actor	Usuario Anónimo
Objetivo	Conocer el día a día del Taller
Descripción	El Usuario puede conocer el trabajo diario del Taller mediante los enlaces disponibles en la parte pública relativos a las últimas entradas del blog de encuentro del Taller.

Caso de Uso	Acceder Información TMB
Actor	Usuario Anónimo
Objetivo	Visualizar la información comercial del Taller
Descripción	El usuario encontrará amplia información acerca de la actividad del Taller en el pasado, presente y futuro. Es un enlace a la presentación institucional de TMB.

Caso de Uso	Visualizar Avisos Legales
Actor	Usuario Anónimo
Objetivo	Conocer los derechos y deberes del Portal TMB
Descripción	El usuario tiene acceso a las líneas legales tanto del diseño como de la utilización del Portal.

Caso de Uso	Autentificarse
Actor	Administrador
Objetivo	Acceder al panel de administración
Descripción	El Administrador accede a los archivos arquitectónicos del portal situados en el servidor del proveedor de alojamiento, a través de claves proporcionadas por este último.

Caso de Uso	Actualizar Noticias
Actor	Administrador
Objetivo	Actualizar las noticias del Mundo Motor
Descripción	El administrador modifica los archivos relativos a las noticias del automóvil, de forma que las entradas sean lo más actuales posibles. Todo ello dentro del servidor del proveedor.

Caso de Uso	Actualizar Consejos de Mantenimiento
Actor	Administrador
Objetivo	Actualizar los consejos de mantenimiento
Descripción	El administrador modifica los archivos relativos a las noticias de mantenimiento del vehículo, de forma que las entradas sean lo más actuales posibles. Todo ello dentro del servidor del proveedor.

Caso de Uso	Actualizar Ofertas Personalizadas
Actor	Administrador
Objetivo	Actualizar las ofertas de los usuarios registrados
Descripción	El administrador modifica las ofertas a las que pueden optar los usuarios registrado, de forma que las ofertas estén siempre vigentes.

Caso de Uso	Actualizar entradas Blog
Actor	Administrador
Objetivo	Actualizar los enlaces del blog de encuentro
Descripción	El administrador modifica los archivos relativos a las entradas del blog del Taller, de forma que las entradas sean lo más actuales posibles. Todo ello dentro del servidor del proveedor.

Caso de Uso	Actualizar Consejos de Seguridad
Actor	Administrador
Objetivo	Actualizar los consejos de seguridad del automóvil
Descripción	El administrador modifica los archivos relativos a las noticias de consejos de seguridad del automóvil, de forma que las entradas sean lo más actuales posibles. Todo ello dentro del servidor del proveedor.

Caso de Uso	Cerrar Sesión
Actor	Administrador
Objetivo	Abandonar de forma segura el panel de control
Descripción	Como medida de seguridad el administrador debe cerrar su sesión en el panel de control.

3.3.3 Diseño del Sistema de Información

Una vez efectuada la propuesta y el posterior análisis de requisitos y casos de uso, se procede a realizar el diseño del Portal Web de TMB.

3.3.3.1 Diseño Arquitectónico

Se utilizará un único servidor en la que se instalará el Almacén de Datos con la información de las reparaciones de los vehículos, los usuarios accederán a la información desde sus dispositivos a través de la red. Los archivos necesarios para el funcionamiento del Portal también estarán almacenados en el servidor, donde identificarán a los usuarios registrados, mediante las claves proporcionadas por TMB, para poder acceder a su zona privada.

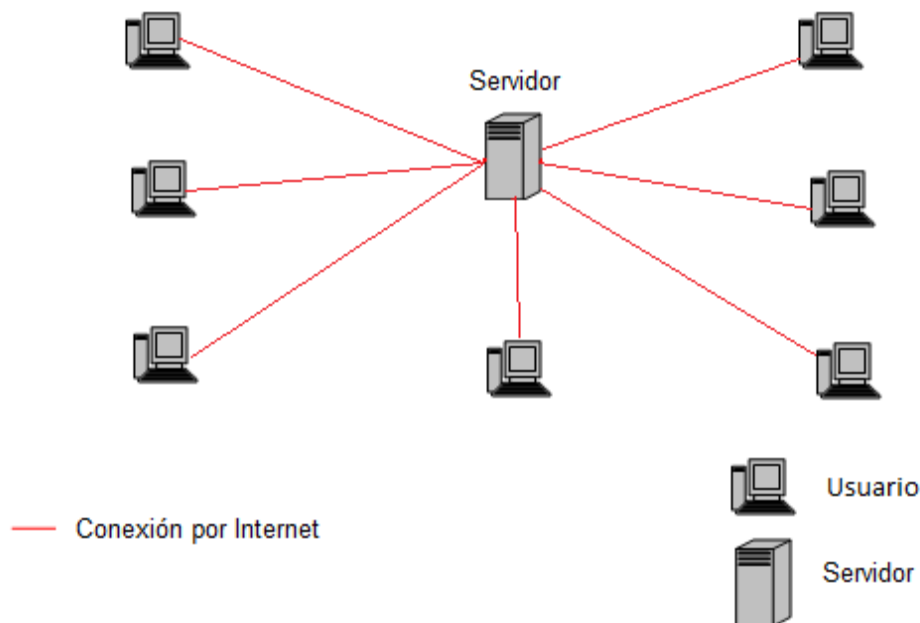


Ilustración 62 Conexión Servidor – Usuarios

El servidor debe ser lo bastante potente para “*correr*” la aplicación de forma regular, por lo que debe ser privado. Los Datos no “*pesan*” demasiado y al ser de sólo lectura el intercambio de información no es excesiva. Los costes en este sentido serán mínimos.

3.3.3.2 Almacén de Datos

Hay que tener en cuenta que la Base de Datos completa se encuentra en local. La aplicación de escritorio, descrita anteriormente, es quien trabaja y actualiza todos los datos del taller.

Para el Libro de Mantenimiento Online, solo necesitaremos los datos referentes a las reparaciones de los distintos vehículos de los clientes. En el **Anexo 2** describimos como extraemos esta información de la Base de Datos *Madre*, como la transformamos, optimizamos e implantamos en el entorno de operación.

Los datos serán consistentes siempre, por los que nos encontramos ante tablas sin integridad referencial que debemos analizar y explicar.

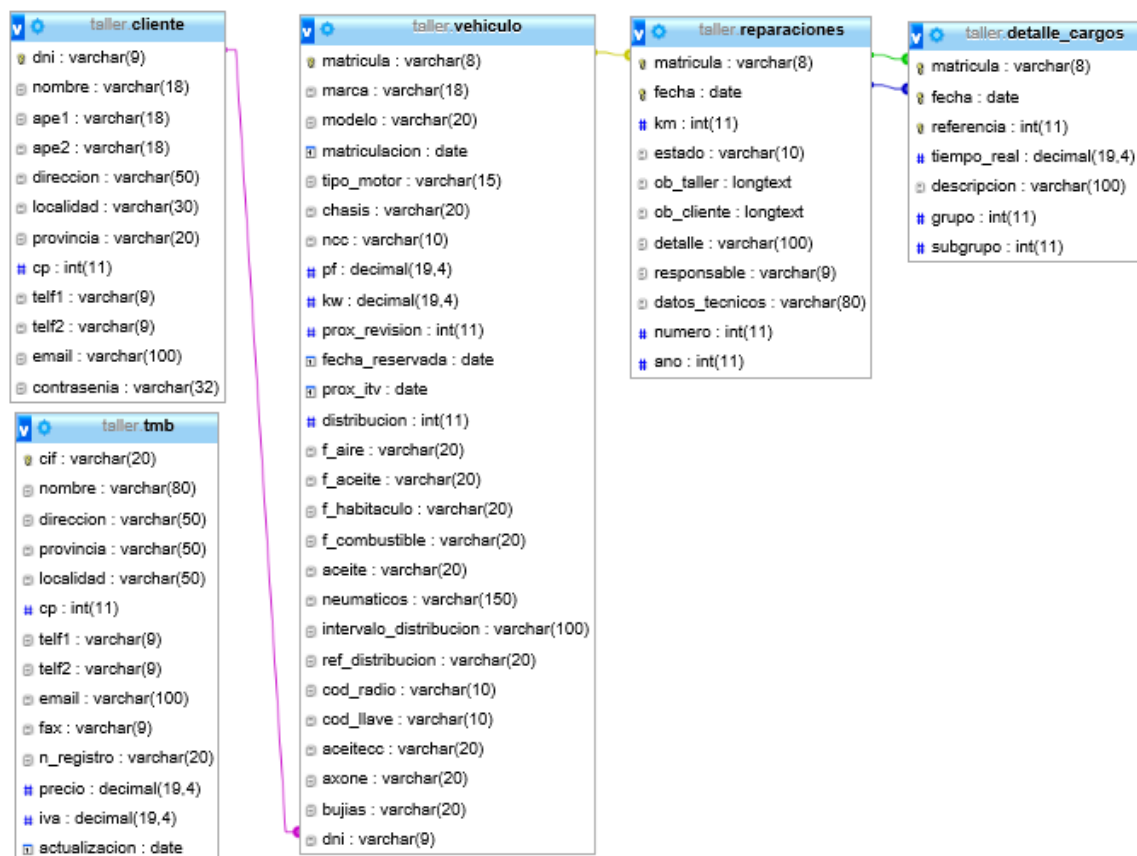


Ilustración 63 Almacén de Datos Portal Web

Merece la pena recordar que esta es la versión alfa del software, y el cliente quiere ver cómo funciona el producto para invertir más en él. Debido a esto, la Base de Datos es sólo de lectura, el usuario solo visualiza el detalle de sus reparaciones.

Descripción de las tablas

Tabla tmb. Recoge información de situación y contacto de la empresa. También almacena la fecha de la última actualización de la Base de Datos para el Portal Web.

Campo	Tipo	Nulo
cif	varchar(20)	No
nombre	varchar(80)	Sí
direccion	varchar(50)	Sí

Campo	Tipo	Nulo
provincia	varchar(50)	Sí
localidad	varchar(50)	Sí
cp	int(11)	Sí
telf1	varchar(9)	Sí
telf2	varchar(9)	Sí
email	varchar(100)	Sí
fax	varchar(9)	Sí
n_registro	varchar(20)	Sí
precio	decimal(19,4)	Sí
iva	decimal(19,4)	Sí
actualizacion	date	Sí

Tabla cliente. Almacena la información personal del cliente, destacando la contraseña de acceso a la zona privada.

Campo	Tipo	Nulo
dni	varchar(9)	No
nombre	varchar(18)	Sí
ape1	varchar(18)	Sí
ape2	varchar(18)	Sí
direccion	varchar(50)	Sí
localidad	varchar(30)	Sí
provincia	varchar(20)	Sí
cp	int(11)	Sí

Campo	Tipo	Nulo
telf1	varchar(9)	Sí
telf2	varchar(9)	Sí
email	varchar(100)	Sí
contrasenia	varchar(32)	No

Tabla vehiculo. Almacena todos los datos técnicos del vehículo así como las fechas más relevantes a la hora de llevar el mantenimiento de un vehículo.

Campo	Tipo	Nulo	Enlaces a
matricula	varchar(8)	No	
marca	varchar(18)	Sí	
modelo	varchar(20)	Sí	
matriculacion	date	Sí	
tipo_motor	varchar(15)	Sí	
chasis	varchar(20)	Sí	
ncc	varchar(10)	Sí	
pf	decimal(19,4)	Sí	
kw	decimal(19,4)	Sí	
prox_revision	int(11)	Sí	
fecha_reservada	date	Sí	
prox_itv	date	Sí	
distribucion	int(11)	Sí	
f_aire	varchar(20)	Sí	
f_aceite	varchar(20)	Sí	

Campo	Tipo	Nulo	Enlaces a
f_habitaculo	varchar(20)	Sí	
f_combustible	varchar(20)	Sí	
aceite	varchar(20)	Sí	
neumaticos	varchar(150)	Sí	
intervalo_distribucion	varchar(100)	Sí	
ref_distribucion	varchar(20)	Sí	
cod_radio	varchar(10)	Sí	
cod_llave	varchar(10)	Sí	
aceitecc	varchar(20)	Sí	
axone	varchar(20)	Sí	
bujias	varchar(20)	Sí	
dni	varchar(9)	No	cliente -> dni

Tabla reparaciones. Toda reparación a almacenar debe contener la información relativa al momento en que se realiza la operación, no solo datos técnicos sino también observaciones.

Columna	Tipo	Nulo	Enlaces a
matricula	varchar(8)	No	vehiculo -> matricula
fecha	date	No	
km	int(11)	Sí	
estado	varchar(10)	Sí	
ob_taller	longtext	Sí	

Columna	Tipo	Nulo	Enlaces a
ob_cliente	longtext	Sí	
detalle	varchar(100)	Sí	
responsable	varchar(9)	Sí	
datos_tecnicos	varchar(80)	Sí	
numero	int(11)	Sí	
ano	int(11)	Sí	

Tabla detalle_cargos. Recoge información de todas las operaciones que se han realizado en la reparación seleccionada.

Campo	Tipo	Nulo	Enlaces a
matricula	varchar(8)	No	reparaciones -> matricula
fecha	date	No	reparaciones -> fecha
referencia	int(11)	No	
tiempo_real	decimal(19,4)	Sí	
descripcion	varchar(100)	Sí	
grupo	int(11)	Sí	
subgrupo	int(11)	Sí	

3.3.3.3 Diseño de la Interfaz de Usuario

El sistema busca que el usuario puede acceder a su libro de mantenimiento desde cualquier dispositivo, y que sea a su vez fácil de recorrer por todos sus niveles.

Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la Web, la forma, organización y flexibilidad del diseño es crítica. Con **HTML5** lograremos nuestro objetivo.

La Portada de la Aplicación

La parte superior de la Web debe ser la más importante y llamativa. En ella encontraremos el acceso a la zona privada para los clientes con claves y el enlace al formulario de contacto con el Taller. El diseño también deja claro que nos encontramos con una aplicación *responsive*. Por último cuatro secciones con imágenes acotan esta primera parte del Portal, son las noticias del Mundo del Motor, de Mantenimiento del vehículo, consejos de seguridad al volante y el enlace directo a la DGT.



Ilustración 64 Parte superior de la portada

Tras esta primera visión de la Web, dedicamos un espacio a conocer la historia del Taller. El típico *sobre nosotros* unido a la metodología de trabajo de la empresa están visibles. Aquí también destacan los enlaces a las últimas entradas del blog de encuentro **Diagnóstico Avería**.

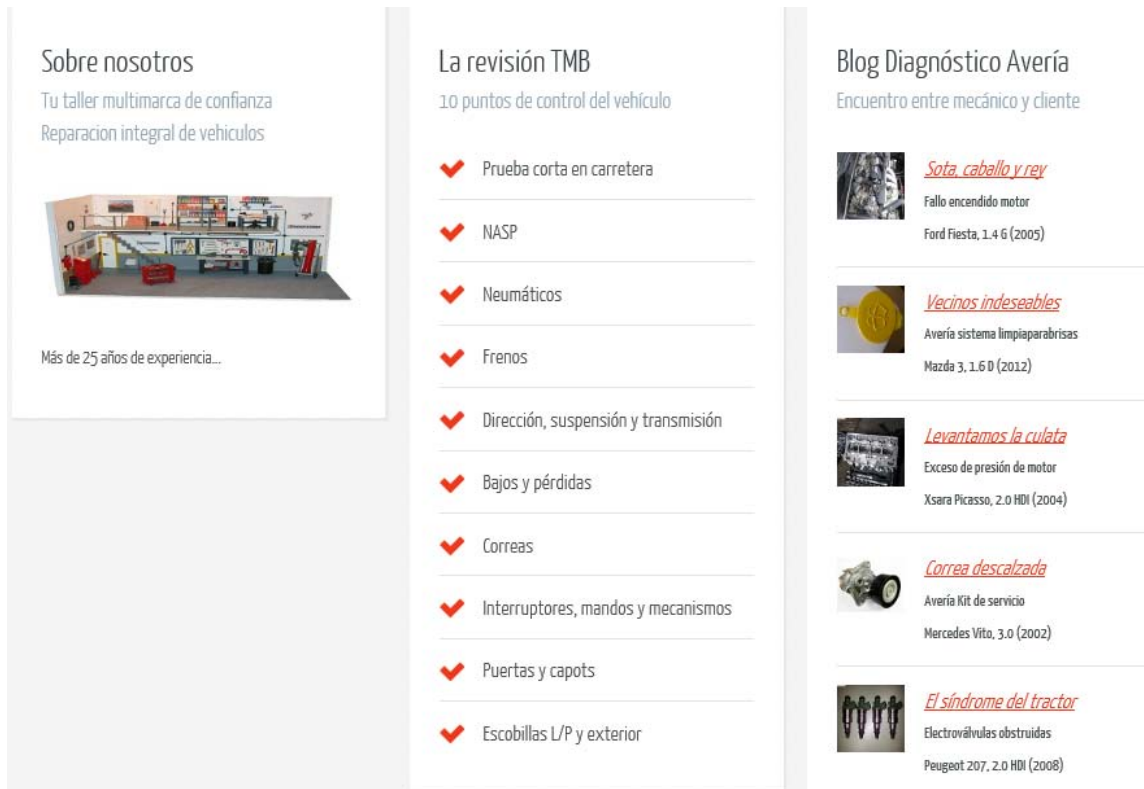


Ilustración 65 Parte central de la portada

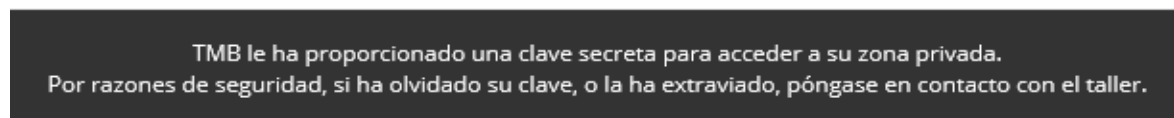
El pie de página de la portada está dedicado a los enlaces de contacto con el Taller, a las redes sociales y avisos legales de diseño.



Ilustración 66 Pie de página de la portada

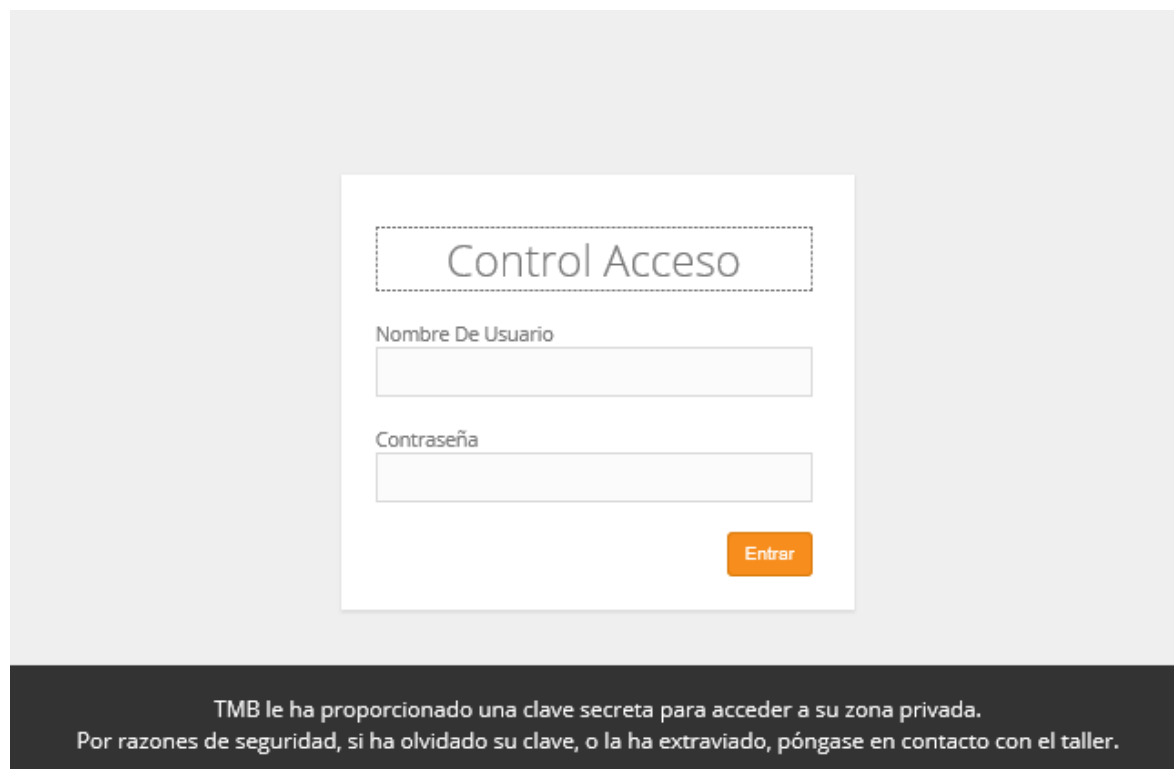
Cabecera y Pie de página

Los siguientes niveles de la Web siguen el patrón marcado por HTML5, donde apreciamos en todo momento una cabecera y un pie de página personalizados.



La pasarela de autenticación

Para acceder a su zona privada el cliente debe autenticarse mediante claves dadas por el Taller. Un formulario de control preside este nivel, sensible a mayúsculas y minúsculas, además de a caracteres especiales.

A screenshot of a web page. The background is light gray. In the center is a white rectangular box containing a login form. At the top of the box is a dashed-line rectangle with the text "Control Acceso". Below this are two input fields: the first is labeled "Nombre De Usuario" and the second is labeled "Contraseña". To the right of the "Contraseña" field is an orange button with the white text "Entrar". Below the white box, there is a dark gray banner with white text that reads: "TMB le ha proporcionado una clave secreta para acceder a su zona privada. Por razones de seguridad, si ha olvidado su clave, o la ha extraviado, póngase en contacto con el taller."

Inicio Zona privada

Nada más entrar en su área privada el usuario puede ver un menú para navegar por los niveles del Portal, un listado con sus vehículos y un catálogo interactivo con sus ofertas personalizadas y noticias que le pueden interesar. También es visible en todo momento el enlace de fin de sesión y la fecha de la última actualización de sus datos.

[Inicio](#) [Aviso Legal](#) [Contacto](#) Actualización: 27/06/2015

área privada
selecciona tu vehículo

MATRÍCULA	MARCA	MODELO	MATRICULACIÓN
4293DSF	FIAT	PANDA	04/11/2005
8078GEV	SEAT	ALTEA XL	09/11/2007
8610GPB	PEUGEOT	308 SPORT HDI	05/08/2009

Revisa tus ofertas y noticias

TE LLEVAMOS
EL COCHE A LA

itv

TOTALMENTE
GRATIS

itv

>

REVISION PRE-ITV INCLUIDA, TASAS DE LA ITV NO INCLUIDAS

Reparaciones de los vehículos

Tras elegir el vehículo a visualizar, el usuario registrado podrá ver las reparaciones que se han realizado al coche seleccionado, así como los kilómetros y la fecha en que se realizó. Cada reparación tiene un título descriptivo para que sea más intuitiva la navegación.

[Inicio](#) [Aviso Legal](#) [Contacto](#)

área privada
selecciona la reparación



Matrícula 8610GPB
Próxima Revisión 107500 Km
Próxima ITV 20/09/2015
Fecha Reservada 21/03/2016

FECHA	KILÓMETROS	DETALLE REPARACIÓN
07/04/2015	92120	NEUMATICOS
31/03/2015	92120	RV/SF.1ª

Detalle de la reparación

Seleccionada la reparación, el detalle de todos los trabajos realizados en esa fecha aparece en pantalla. Están ordenados por grupos de trabajo y orden de realización.

[Inicio](#) [Índice Reparaciones](#) [Aviso Legal](#) [Contacto](#)

área privada
consulta la reparación



Matrícula	8610GPB
Fecha Reparación	31/03/2015
Km Reparación	92120

DESCRIPCIÓN DE LA REPARACIÓN

RV TMB
PREPARACIÓN DEL TRABAJO
AUTODIAGNOSIS DE SISTEMAS Y RESET
SUSTITUIR GRAPA VARILLA DE CAPOT
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE
SUSTITUIR FILTRO DE AIRE
SUSTITUIR FILTRO DE HABITACULO
REPARACION DE FRENOS:
SUSTITUIR KIT DE FRENOS DELANTEROS
SUSTITUIR LIQUIDO DE FRENOS
SUSTITUIR TIRANTA DE SUSPENSION D/D.

Aviso legal

Es importante cubrir legalmente a la empresa, por lo que existe un enlace para que el cliente conozca sus derechos y deberes a la hora de utilizar el servicio TMB.

[Inicio](#) [Aviso Legal](#) [Contacto](#)

Aviso legal y Política de privacidad

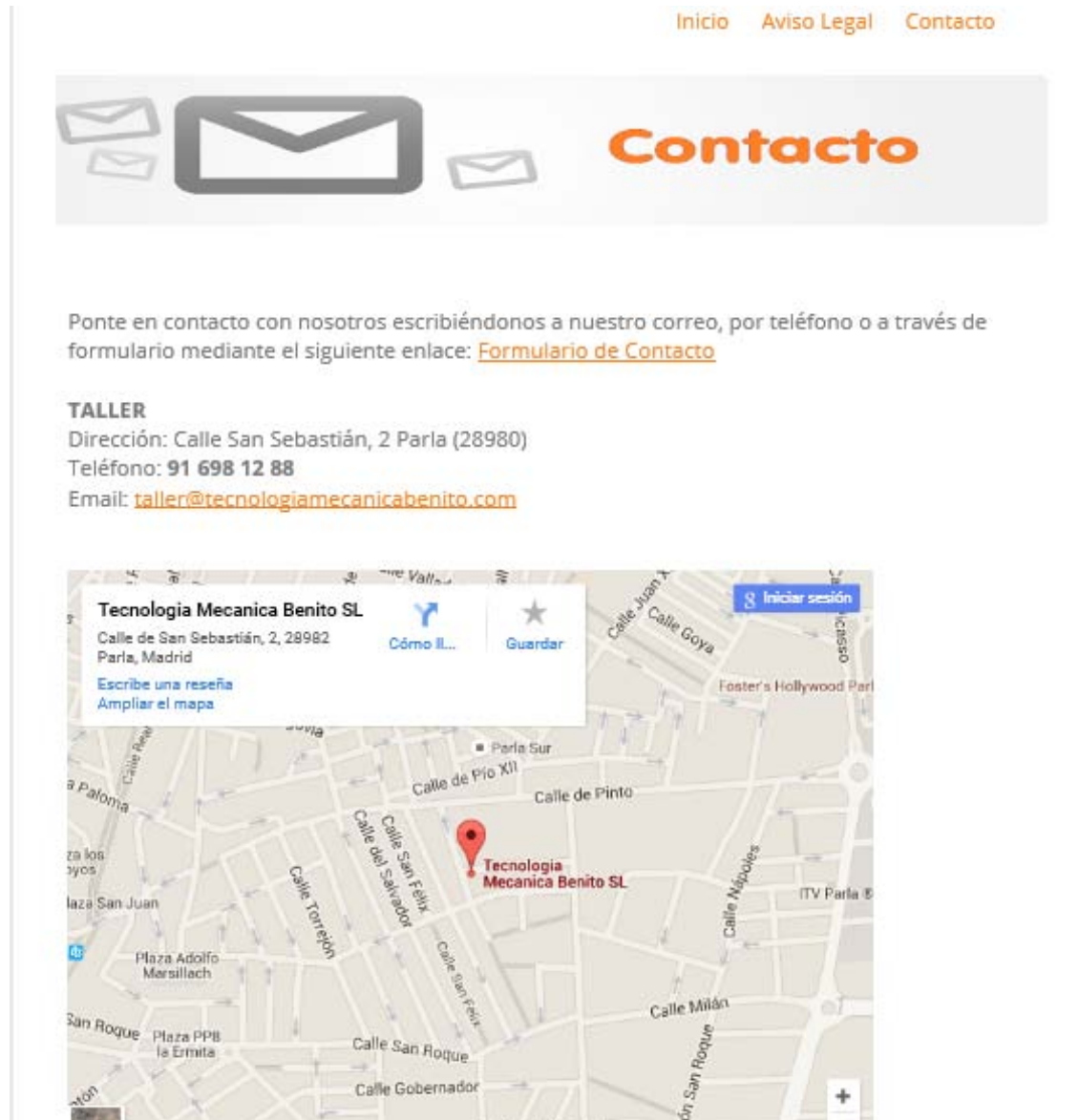
1. DATOS IDENTIFICATIVOS: En cumplimiento con el deber de información recogido en artículo 10 de la Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y del Comercio Electrónico, a continuación se reflejan los siguientes datos: la empresa titular de www.tecnologiamecanicabenito.com es Tecnología Mecánica Benito SL (en adelante TMB), con domicilio a estos efectos en Calle San Sebastián 2, número de C.I.F.: B81370041, con correo electrónico de contacto: taller@tecnologiamecanicabenito.com

2. USUARIOS: El acceso y/o uso de este portal de TMB atribuye la condición de USUARIO, que acepta, desde dicho acceso y/o uso, las Condiciones Generales de Uso aquí reflejadas. Las citadas Condiciones serán de aplicación independientemente de las Condiciones Generales de Contratación que en su caso resulten de obligado cumplimiento.

3. USO DEL PORTAL: www.tecnologiamecanicabenito.com proporciona el acceso a multitud de informaciones, servicios, programas o datos (en adelante, "los contenidos") en Internet pertenecientes a TMB o a sus licenciantes a los que el USUARIO pueda tener acceso. El USUARIO asume la responsabilidad del uso del portal. Dicha responsabilidad se extiende al registro que fuese necesario para acceder a determinados servicios o contenidos. En dicho registro el USUARIO será responsable de aportar información veraz y lícita. Como consecuencia de este registro, al USUARIO se le puede proporcionar una contraseña de la que será responsable, comprometiéndose a hacer un uso diligente y confidencial de la misma. El USUARIO se compromete a hacer un uso adecuado de los contenidos y servicios (como por ejemplo servicios de chat, foros de discusión o grupos de noticias) que TMB ofrece a través de su portal y con carácter enunciativo pero no limitativo, a no emplearlos para (i) incurrir en actividades ilícitas, ilegales o contrarias a la buena fe y al orden público; (ii) difundir contenidos o propaganda de carácter racista, xenófobo, pornográfico-ilegal, de apología del terrorismo o atentatorio contra los derechos humanos; (iii) provocar daños en los sistemas físicos y lógicos de TMB, de sus proveedores o de terceras personas, introducir o difundir en la red virus informáticos o cualesquiera otros sistemas físicos o lógicos que sean susceptibles de provocar los daños

Contacto

Es importante que el cliente interactúe con el Taller. Tiene la opción de ponerse en contacto con la empresa mediante teléfono, mail y formulario. También tiene acceso a un pequeño mapa interactivo.



Formulario de Contacto

Contacta con nosotros

Ponte en contacto con nosotros a través de este formulario, por teléfono o escribiéndonos a nuestro correo.

Dirección: Calle San Sebastián, 2 Parla (28980)
Teléfono: 91 698 12 88
Correo: taller@tecnologiamecanicabenito.com

***Obligatorio**

Nombre *


Teléfono *

Email *

Mensaje *

Enviar

Nunca envíes contraseñas a través de Formularios de Google.

Con la tecnología de  Google Forms

Este formulario se creó en TECNOLOGIA MECANICA BENITO S.L. - [Informar sobre abusos](#) - [Condiciones del servicio](#) - [Otros términos](#)

Utilizamos los recursos de google para contactar con nuestros clientes. De esta forma evitamos ataques y correo no deseado.

3.3.4 Implementación del Sistema de Información

Con el diseño de la Base de Datos y el Interfaz de Usuario ya podemos construir el sistema de información. En este capítulo también haremos referencia a la implantación del Portal Web en el entorno de operación.

3.3.4.1 Construcción del Sistema Información

Para la construcción del Portal es necesario conocer el entorno de desarrollo de la aplicación y como se ejecutará en el entorno de operación.

3.3.4.1.1 Entorno de Desarrollo

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

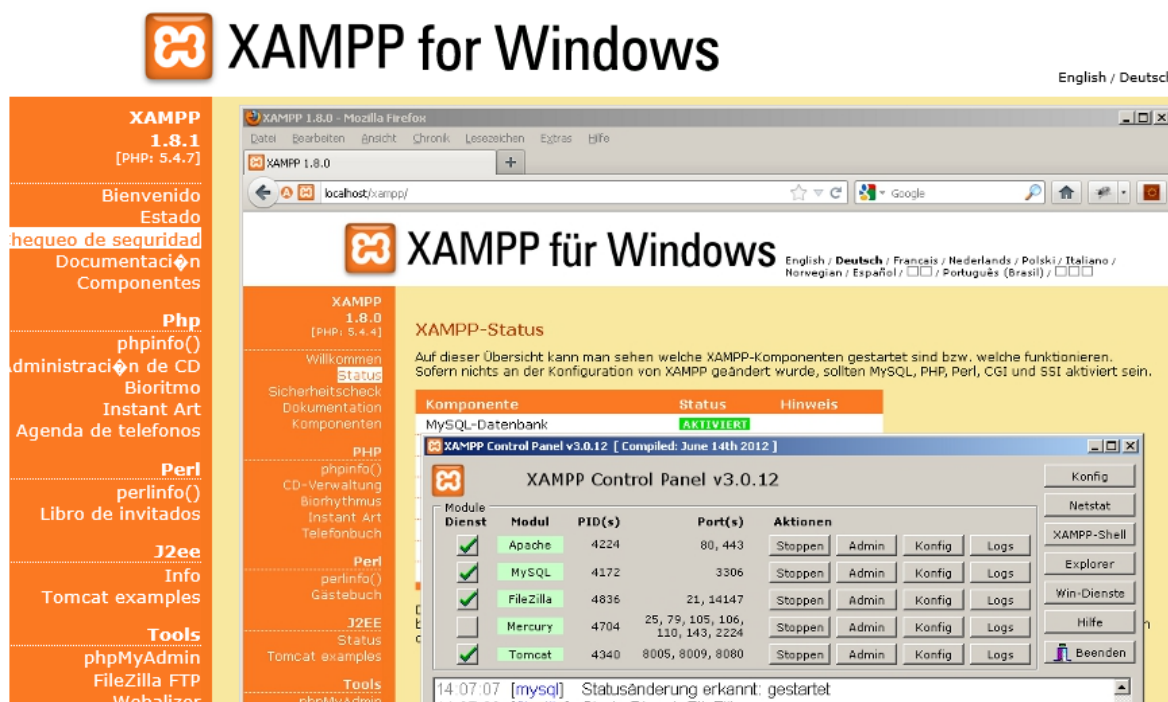


Ilustración 67 XAMPP Plataforma de Desarrollo

Gracias a esta plataforma montaremos nuestro servidor en local, capaz de interpretar PHP, lenguaje con el que trataremos nuestra Base de Datos. Al tener el servidor en local las pruebas podrán hacerse al instante de desarrollar el código.

```

7 <html>
8 <head>
9     <title>tecnologiamecanicabenito.com</title>
10    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
11    <meta name="description" content="" />
12    <meta name="keywords" content="" />
13    <script src="js/jquery.min.js"></script>
14    <script src="js/skel.min.js"></script>
15    <script src="js/skel-layers.min.js"></script>
16    <script src="js/init.js"></script>
17    <noscript>
18        <link rel="stylesheet" href="css/skel.css" />
19        <link rel="stylesheet" href="css/style.css" />
20        <link rel="stylesheet" href="css/style-desktop.css" />
21    </noscript>
22    <!--[if lte IE 9]><link rel="stylesheet" href="css/ie9.css" /><![endif]-->
23    <!--[if lte IE 8]><script src="js/html5shiv.js"></script><![endif]-->
24 </head>
25 <body>
26
27     <!-- Header -->
28     <div id="header-wrapper">
29         <header id="header" class="container">
30             <div class="row">
31                 <div class="12u">

```

Ilustración 68 Notepad++ Editor de código

Para el desarrollo del código utilizamos **Notepad++**, un editor de texto y de código fuente libre con soporte para PHP. Sin olvidar **phpMyAdmin**, que gestionará nuestra Base de Datos y será crucial a la hora de la transformación y migración de los datos al servidor final. (ANEXO 2).

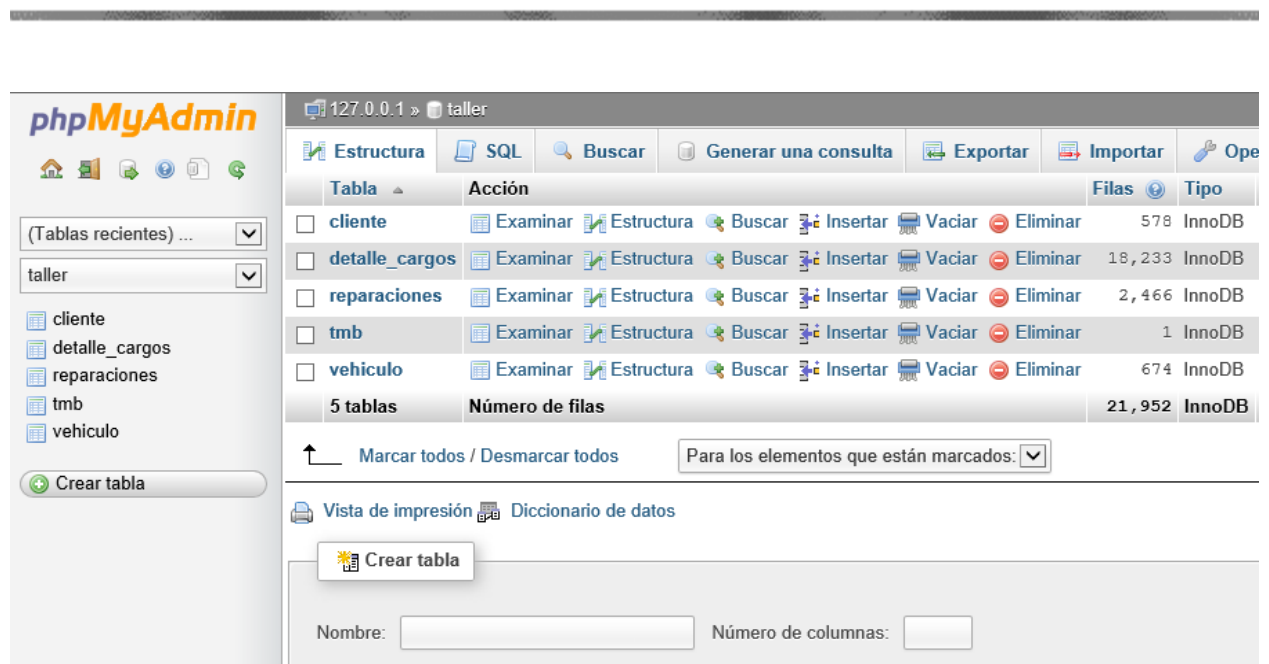


Ilustración 69 Herramienta phpMyAdmin

Esta herramienta está escrita en PHP, capaz de manejar la administración de MySQL a través de páginas web. Puede crear y eliminar BBDD, manejar tablas, campos y ejecutar cualquier sentencia SQL. También es capaz de administrar claves, privilegios y exportar datos en varios formatos.

3.3.4.1.2 Entorno de Ejecución

PHP es un lenguaje de programación de código del lado del servidor. El código es interpretado por un servidor Web con un módulo de procesador de PHP que genera la página Web resultante.

El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP.

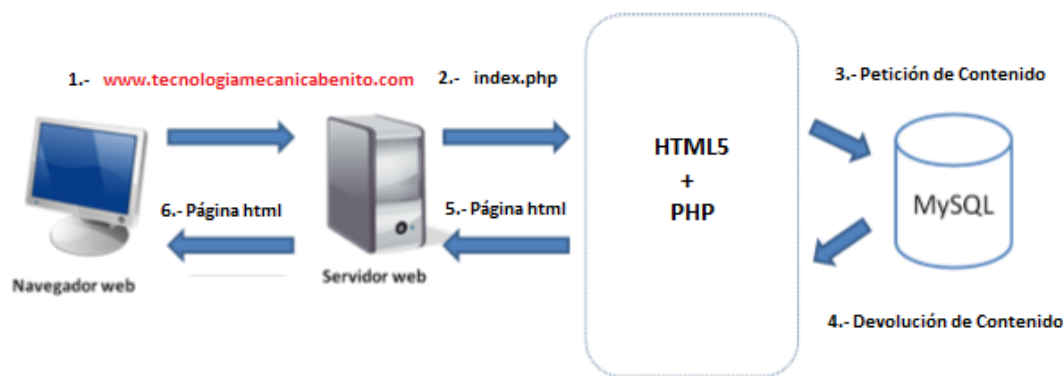


Ilustración 70 Entorno de Ejecución Portal Web

- 1.- Se introduce la dirección web de la página que se quiere visualizar, <http://www.tecnologiamecanicabenito.com/>, y el navegador realiza la solicitud al servidor web.
- 2.- El archivo index.php es el esqueleto del portal web. El servidor añadirá contenido a partir de este archivo. Se interpretará PHP para crear la página index.php a devolver, incluyendo módulos y contenido necesario, aplicando el estilo de la plantilla actual. (HTML5)
- 3.- Se establece una conexión con la Base de Datos y le solicita el contenido necesario para servir la página.
- 4.- La Base de Datos devuelve el contenido solicitado.
- 5.- Se devuelve el contenido en forma de HTML al navegador web, a través del servidor.
- 6.- El navegador web se encarga de mostrar el contenido de la página al usuario.

3.3.4.1.3 Codificación de los Niveles del Portal

Diseñada la Interfaz de Usuario vamos analizar los aspectos de codificación más importantes de los distintos niveles de la Web.

La Portada de la Aplicación

La pantalla de inicio de la aplicación va a ser visitada tanto por los usuarios registrados como por los anónimos. Debe ser lo más atractiva posible ya que no sólo va a ser representativa de la

imagen de la empresa, también debe enganchar a nuevos clientes. Al ser un elemento tan cuidado y que debe visualizarse en los distintos dispositivos, se va optar por una plantilla *responsive* 100% personalizable para la portada, a partir de la cual montaremos en niveles ocultos la zona privada de los clientes.

La codificación de la plantilla se encargará de detectar ante que dispositivo nos encontramos y redimensionar la Web en consecuencia.

```

1  <!DOCTYPE HTML>
2  <!--
3      Halcyonic by HTML5 UP
4      html5up.net | @n33co
5      Free for personal and commercial use under the CCA 3.0 license (html5up.net/license)
6  -->
7  <html>
8      <head>
9          <title>tecnologiamecanicabenito.com</title>
10         <meta http-equiv="content-type" content="text/html; charset=utf-8" />
11         <meta name="description" content="" />
12         <meta name="keywords" content="" />
13         <script src="js/jquery.min.js"></script>
14         <script src="js/skel.min.js"></script>
15         <script src="js/skel-layers.min.js"></script>
16         <script src="js/init.js"></script>
17         <noscript>
18             <link rel="stylesheet" href="css/skel.css" />
19             <link rel="stylesheet" href="css/style.css" />
20             <link rel="stylesheet" href="css/style-desktop.css" />
21         </noscript>
22         <!--[if lte IE 9]><link rel="stylesheet" href="css/ie9.css" /><![endif]-->
23         <!--[if lte IE 8]><script src="js/html5shiv.js"></script><![endif]-->
24     </head>
25     <body>

```

Ilustración 71 Plantilla *Responsive*

Cabecera y Pie de página

Los niveles ocultos seguirán las indicaciones de HTML5. La cabecera y el pie de página se codificarán en archivos PHP distintos, llamados desde el bloque de artículos para construir el HTML a visualizar.

Tecnología Mecánica Benito

Bienvenido al servicio de taller on line de TMB

TMB le ha proporcionado una clave secreta para acceder a su zona privada.

Por razones de seguridad, si ha olvidado su clave, o la ha extraviado, póngase en contacto con el taller.


```

65 $clave_busqueda=$_SESSION['session_username'];
66 $query = "select matricula,marca,modelo,date_format(matriculacion, '%d/%m/%Y') as matr
67 $result = mysql_query($query);
68 $variable=0;
69 while($fila=mysql_fetch_array ($result)){
70     ?>
71     <tr id="fila_<?php echo $variable ?>" onMouseOver="ResaltarFila('fila_<?php echo $vari
72     <td width='150'><a href="zonaprivada2.php?pasa_vehiculo=<?php echo $fila['matricula']?
73     <td width='150'><?php echo $fila['marca']; ?></td>
74     <td width='150'><?php echo $fila['modelo']; ?></td>
75     <td width='150'><?php echo $fila['matriculacion'];?></td>
76     </tr>
77     <?php
78     $variable++;
79     } ?>
80     <?php
81     $numero1 = mysql_num_rows($result);
82     //echo"<thead><th width='150' colspan='7' rowspan='7' align='center'>El Total de V
83     ?>
84     </table>
85 </p>
86 <?php
87     include("includes/menu.html"); ?>
88     </article> <!-- /article -->
89
90 </section> <!-- / #main-content -->
91 </div>

```

Ilustración 72 Llamada a pie de página

La pasarela de autenticación

Es el punto de crítico en cuestión de seguridad. El cliente debe introducir las claves de acceso proporcionadas por TMB y ahí está el peligro. Evitamos la inyección de SQL y JavaScript (ataques Xss) dando sensibilidad a las cajas de texto para que el sistema distinga entre mayúsculas y minúsculas, no permita caracteres especiales y acote el número de letras a introducir.

```

1  <?php
2  session_start();
3  ?>
4
5  <?php require_once("includes/connection.php"); ?>
6  <?php include("includes/header.php"); ?>
7
8
9
10 <?php
11
12 if(isset($_SESSION["session_username"])){
13     // echo "Session is set"; // for testing purposes
14     header("Location: intropage.php");
15 }
16
17 if(isset($_POST["login"])){
18
19     if(!empty($_POST['username']) && !empty($_POST['password']) && preg_match("/^[0-9a-zA-Z]+$/", $_POST['username']) && preg_match("/^[0-9a-zA-Z]+$/",
20         $username=strtoupper($_POST['username']));
21         $password=strtoupper($_POST['password']));
22
23         // $query =mysql_query("SELECT * FROM CLIENTE WHERE dni='".$_.$username."' AND clave='".$_.$password."'");
24         $query =mysql_query("SELECT * FROM CLIENTE WHERE dni='".$_.$username."'");
25
26         $numrows=mysql_num_rows($query);
27         if($numrows!=0)
28

```

Ilustración 73 Cajas de Texto sensibles

Las contraseñas que se proporcionan al usuario están encriptadas con MD5 en la Base de Datos. El proceso se realiza en la migración de Access a MySQL mediante script.

```
ALTER TABLE `cliente` ADD `contrasenia` VARCHAR(32);
```

```
UPDATE `cliente` SET `contrasenia`=MD5(`clave`);
```

Ámbito Privado

La estructura HTML5 preside esta parte de la aplicación.

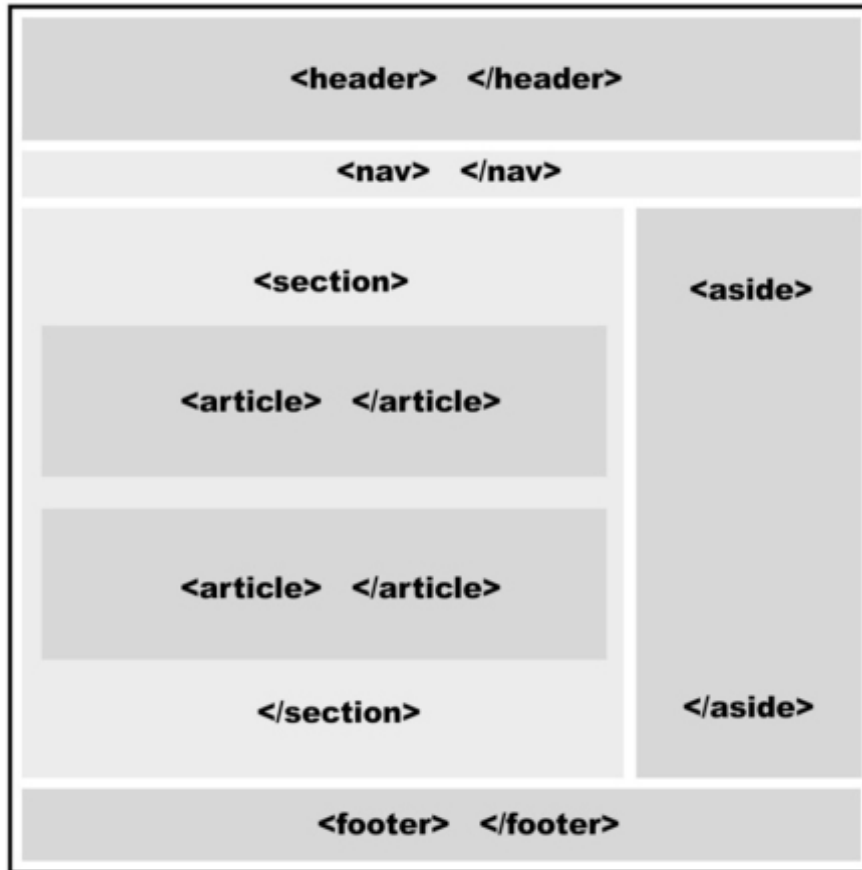


Ilustración 74 Estructura HTML5

Utilizamos un poco de javascript para dar movimiento a las tablas que muestran los distintos vehículos, con sus reparaciones y el detalle de estas. Son tablas tipo cebra que se *iluminan* cuando pasan el ratón por las distintas filas. Dentro de las celdas hay enlaces para desplazarnos a los distintos vehículos, reparaciones y detalles de los trabajos realizados.

```
22 <script language="javascript" type="text/javascript">
23 // RESALTAR LAS FILAS AL PASAR EL MOUSE
24 function ResaltarFila(id_fila) {
25     document.getElementById(id_fila).style.backgroundColor = '#C0C0C0';
26 }
27
28 // RESTABLECER EL FONDO DE LAS FILAS AL QUITAR EL FOCO
29 function RestablecerFila(id_fila) {
30     document.getElementById(id_fila).style.backgroundColor = '#FFFFFF';
31 }
```

Ilustración 75 Código javascript

```

<table id="tabla_vehiculo" align="center" border="0" cellpadding="2" cellspacing="2">
  <thead><th colspan="1" rowspan="1" align="left">FECHA</th> <th colspan="1" rowspan="1" align="left">KIL&acute;METROS</th>
<?php
$query = "select date_format(fecha, '%d/%m/%Y') as fecha_reparacion,km,detalle from reparaciones where matricula='".$clave_bu;
$result = mysql_query($query);
$variable=0;
while($fila=mysql_fetch_array ($result)){
  <?
  <tr id="fila <?php echo $variable ?>" onMouseOver="ResaltarFila('fila_<?php echo $variable ?>');" onMouseOut="RestablecerFila
  <td width='150'><a href="zonaprivada3.php?pasa_fecha=<?php echo $fila['fecha_reparacion']?>&pasa_vehiculo=<?php echo $clave_bu
  <td width='150'><?php echo $fila['km']; ?></td>
  <td width='150'><?php echo $fila['detalle']; ?></td>
</tr>
  <?php
$variable++;
} ?>
<?php

```

Ilustración 76 Tabla Cebra

Formulario de Contacto

Si bien utilizamos los recursos de google para comunicarnos con el cliente, también damos un toque personalizado al formulario, añadimos código para que además de recibir la empresa el correo, el usuario reciba un acuse de recibo indicándole que su petición será respondida en breve.

```

Código.gs
14 // emailBody para aquellos equipos que no procesan HTML, es texto plano
15 var emailBody = "Correo de contacto de " + name +
16               "\nTexto: " + texto +
17               "\nCon email: " + mail +
18               "\nCon telefono: " + telefono +
19               "\nregistrado: " + timestamp;
20
21 // Para equipos que si procesan HTML
22 // podemos agregar imagenes y cualquier cosa escrita en HTML
23 var htmlBody = "Registro Email de <font color=\"red\"><strong>" + name + "</strong></font>" + ", " +
24               "<br/><br/>La informacion es la siguiente: " +
25               "<br/>Nombre: <font color=\"red\"><strong>" + name + "</strong></font>" +
26               "<br/>" + texto +
27               "<br/>Con email: " + mail +
28               "<br/>Con teléfono: " + telefono +
29               "<br/>Su consulta será tratada en breve, Gracias <i>" + timestamp + "</i>" +
30               // Texto Protección de Datos
31               "<br/><br/><br/> Tecnología Mecánica Benito SL" +
32               "<br/>Email: taller@tecnologiamecanicabenito.com" +
33               "<br/>Tel: 91 698 12 88" +
34               "<br/>-----" +
35               "<br/><br/>La información contenida en la presente transmisión es confidencial y su uso únicamente
36               "<br/>Cláusula Informativa (art. 5.1 LOPD): Le informamos que sus datos de contacto electrónico es
37
38
39
40 var advancedOpts = { name: "Confirmación Contacto", htmlBody: htmlBody, cc: "taller@tecnologiamecanicabenito.com"};
41 // Esta instruccion envia el correo
42 MailApp.sendEmail(mail, subject, emailBody, advancedOpts);
43 }

```

Ilustración 77. Código Añadido Formulario

3.3.4.2 Implantación del Sistema Información

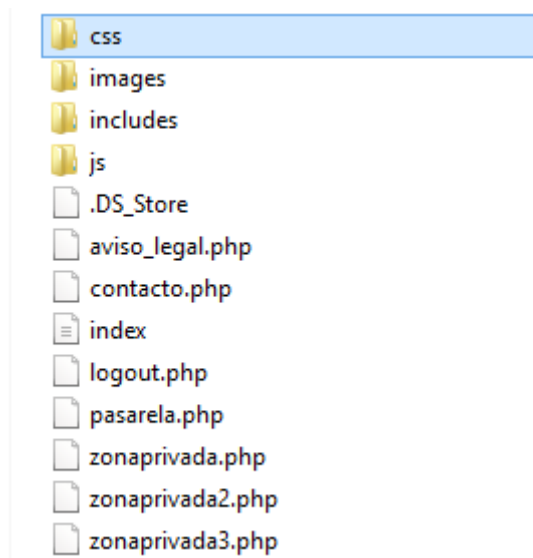
Hay que tener en cuenta que gracias a **XAMPP** satisfacemos los requisitos de hardware y software mínimos, tanto en el lado del desarrollo como en el del usuario final.

Utilizamos software libre, montamos el servidor en local y sabemos que nuestro proveedor de almacenamiento utiliza el mismo gestor de bases de datos que nosotros. Para el Taller eso supone gastos de desarrollo mínimos, y para el Usuario no le supone ningún problema, ya que podrá disponer del servicio desde cualquier dispositivo gracias a HTML5.

Solo queda realizar una buena gestión de contenidos, tanto de formato como de organización, para realizar una implantación en el entorno de operación correcta.

3.3.4.2.1 Organización de Contenidos

Una buena organización de los contenidos sería la siguiente: En el directorio raíz podrían ir las portadas de los distintos niveles de la Web. También en la raíz encontraríamos las distintas carpetas con los script de java, las hojas de estilo, las imágenes y una carpeta *includes*, donde podemos incluir partes de código PHP para construir nuestro Portal, como puede ser la cabecera y el pie de página.



La Base de Datos también estará almacenada en el servidor privado. El proveedor utiliza el mismo gestor de BBDD que nosotros **phpMyAdmin**.

3.3.4.2.2 El entorno de Operación

Carga y migración de datos

Nuestro proveedor de servicios utiliza **phpMyAdmin** para la gestión de las BBDD. La empresa utilizará los backup de sus bases de datos como momento de actualización de los datos del Portal. Hay que tener en cuenta que es un taller de reparaciones que se alargan en el tiempo, por lo que no es necesaria una actualización constante. Todo ello se explica en el **Anexo 5**.

El proceso de migración de los datos está recogido en el **ANEXO 2**. Sólo se transfieren los datos sensibles a las reparaciones de los vehículos, dejando los datos generales del funcionamiento del Taller para futuras actualizaciones del Portal Web. El programa **Bullzip MS Access to MySQL** juega una papel determinante en la conversión de Access a MySQL.

4 Conclusiones y Líneas Futuras

4.1 Conclusiones

El proyecto buscaba satisfacer en primer lugar, la necesidad de **informatizar toda la gestión de los clientes y proveedores**. Conseguir un sistema de información eficiente, que liberara a los empleados cualificados de esa función para dedicarse a la mecánica, además de evitar las inconsistencias de datos que se daban trabajando sólo con papel. En este proceso, interactuar con el administrativo que trabajaría con el programa, fue vital para la programación del interfaz de usuario. No todo quedaba en un buen diseño de la Base de Datos y su conexión con el lenguaje de programación, **aprender a diseñar un interfaz sencillo y eficaz**, para alguien que no fuera experto en automoción, fue igual de importante.

Con el problema de la gestión de oficina resuelto, el siguiente requisito a satisfacer era conseguir la **fidelidad del cliente**. En un entorno de competitividad feroz, y con un mercado de trabajo en continuo cambio, la introducción del **Método Canvas** como modelo de trabajo *dio aire* al taller. Gracias a ser un modelo flexible, de rápida configuración y cambio, la empresa identificó la manera no sólo de llegar a los clientes antiguos, sino de conseguir nuevos clientes.

El Taller se decantó por un Portal Web para captar nuevos clientes, haciéndolo dinámico en su contenido, y ofreciendo algo que no tenían sus competidores cercanos, un libro de mantenimiento online para los clientes antiguos. La empresa buscó la diferenciación respecto a competidores y potenció un servicio personalizado para fidelizar la clientela.

El Portal Web debía ser accesible desde todo tipo de dispositivos, la Web debía adaptarse a los diferentes tamaños y características de los mismos. El acercamiento al **Diseño Web Responsive** era obligatorio para conseguir llegar a todos los clientes. Basándose en redimensionar y colocar los elementos de la Web para adaptarlos a los diferentes dispositivos, el estudio de **HTML5** para completar esta fase fue básico. No se pueden redimensionar los contenidos si no sabes cómo se articulan, **HTML5** respondió a esta pregunta con tres características: **estructura, estilo y funcionalidad**.

Aunque el Portal Web es sólo un prototipo, y aún hay que ver si cumplirá las expectativas, creo que tendrá aceptación entre los clientes gracias al trato personalizado y a la comodidad de consulta de los datos de los vehículos.

4.2 Líneas Futuras

La aplicación de escritorio está completamente operativa, y el Portal Web en fase de aceptación por parte del usuario registrado. Si hay buena acogida por parte de los clientes que visitan el Portal, el paso natural sería ampliar las funcionalidades de la Web, tanto de la parte privada como de la pública, una gestión de **Noticias y Ofertas más detallada y personalizada**, buscando que el usuario interactúe y comente.

El programa de escritorio siempre tiene la posibilidad de mejorar su interfaz de usuario, atendiendo a los comentarios del administrativo. Pero tarde o temprano toda la funcionalidad del programa se trasladará a la Web. Existirá una zona de administrador en el que **la Base de Datos se gestione Online** y el programa de escritorio desaparezca.

El siguiente paso en la búsqueda de la fidelidad del cliente, y que este se sienta identificado con la empresa, es la creación de una **aplicación móvil**, descargable desde su zona privada. Gracias a ella la comunicación entre el cliente y taller existirá en todo momento (Avisos, Consultas, Contacto).

El blog **Diagnostico Avería** deberá ser una realidad. El Taller podrá asociarse con empresas de venta de repuestos online, ya que las entradas del blog contendrán información de averías resueltas con enlaces a las ofertas del proveedor de repuestos Online, llevándose TMB un porcentaje de la venta en caso de compra.

5 Referencias

- [1] “Generación de Modelos de Negocio” por A. Osterwalder
- [2] Artículo “Un negocio en un folio” por Ana Torres - elpais.com
- [3] Práctica “Interfaces Gráficas de Usuario” por N. Prieto

Basado: Weiss. Estructuras de datos en Java. Apéndice D. Ed Addison-Wesley.

- [4] Referencia Internet
- [5] Trabajo “Programación Orientada a Eventos” por A. Molas

Basado: Visual Basic: Los secretos de Visual Basic por Harold Davis.

- [6] “El Libro de Visual Basic 6” por Eric A. Smith, Valor Whisler, Hank Marquis
- [7] PFC “Diseño y desarrollo de un portal Web para un grupo de investigación”

por Lidia María Seldas Alarcón

- [8] Trabajo “Historia del Diseño Web” por Jorge Forero
- [9] “El gran libro de HTML5, CSS3 y Javascript” por Juan Diego Gauchat
- [10] “Guía breve de estándares” - www.w3c.es
- [11] Trabajo “Generaciones del Diseño Web” por Diana Álvarez

6 Anexos

6.1 Método Canvas TMB










Lienzo De Modelo De Negocios		Diseñado para: Tecnología Mecánica Benito	Diseñado por: Luis Marino López	En: <input type="text"/>	Revisión: <input type="text"/>
Socios Clave  <p>Proveedores de repuestos y herramientas. Proveedores locales y casas oficiales.</p>	Actividades Clave  <p>Especialización en reparaciones integrales de vehículos multimarca.</p>	Propuesta de Valor  <p>Buena atención pre y post reparación. Comodidad (Recogida y entrega de vehículos)</p>	Relación con Clientes  <p>Trato rápido y eficaz, garantizando un servicio perfecto pre y post reparación. Que entiendan que hablan con un experto y que son asesorados de forma honesta (Personalización)</p>	Segmentos De Clientes  <p>Amplio mercado: Personas entre 25 y 65 años. Perfil económico medio. Comunidad de Madrid.</p>	
Recursos Clave  <p>Personal: Mecánicos cualificados. Infraestructura: Taller funcional. Tecnología: Siempre con últimos sistemas de diagnóstico. Capital: Bancos (inicial) y reinversión</p>		Canales  <p>WEB RRSS TELÉFONO MAIL</p>			
Estructura De Costos  <p>Nóminas Proveedores Gestoría Préstamos Impuestos</p>			Fuente De Ingresos  <p>El cliente pagará las reparaciones en efectivo, con tarjeta o con transferencia bancaria.</p>		

Ilustración 78. Lienzo de Modelo de Negocio

6.2 Migración Access to MySql

A continuación se presenta una pequeña guía de la migración de los datos a mostrar en el Portal Web. Partiremos de una Base de Datos relacional (**Access**), extraeremos la información, y la almacenaremos en el servidor para mostrarla en la Web (**MySQL**).

En el proceso necesitaremos **Bullzip MS Access to MySQL**, programa para mover el contenido existente en una Base de Datos Access a una Base de Datos MySQL. También será necesaria la herramienta **phpMyAdmin** para manejar la administración de MySQL, sin olvidar la utilización de algún compresor de archivos, en nuestro caso **WinZip**.

El proceso se inicia con la ejecución de **Bullzip MS Access to MySQL**, indicamos la ruta de nuestra Base de Datos y su contraseña en caso de tenerla. Marcamos la opción de creación de un fichero SQL (para la construcción de la Base de Datos MySQL), y elegimos como método de almacenamiento MyISAM en lugar de InnoDB, para “aligerar” el peso del archivo, ya que **phpMyAdmin** tiene un tiempo máximo de importación.

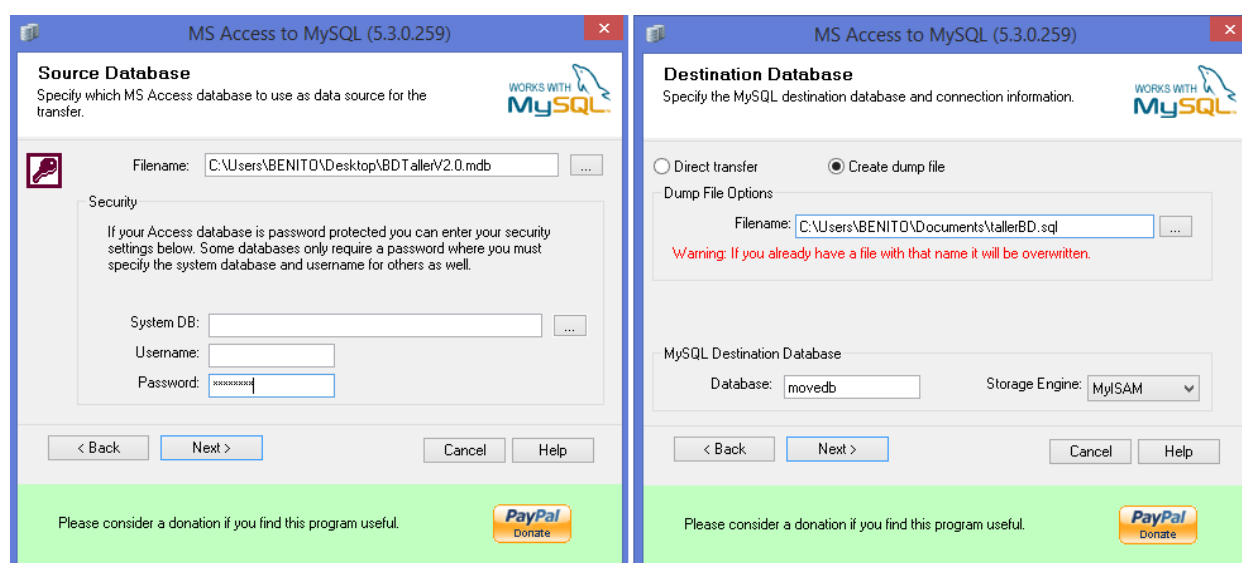


Ilustración 79. MS Access to MySQL

La aplicación nos da la posibilidad de elegir las tablas con las que formar nuestra nueva Base de Datos, en nuestro caso sólo seleccionamos la información relativa a las reparaciones de los vehículos (datos de facturación, almacén y proveedores quedan fuera). Seguimos las instrucciones de ejecución y conseguimos el archivo SQL de la Base de Datos convertida.

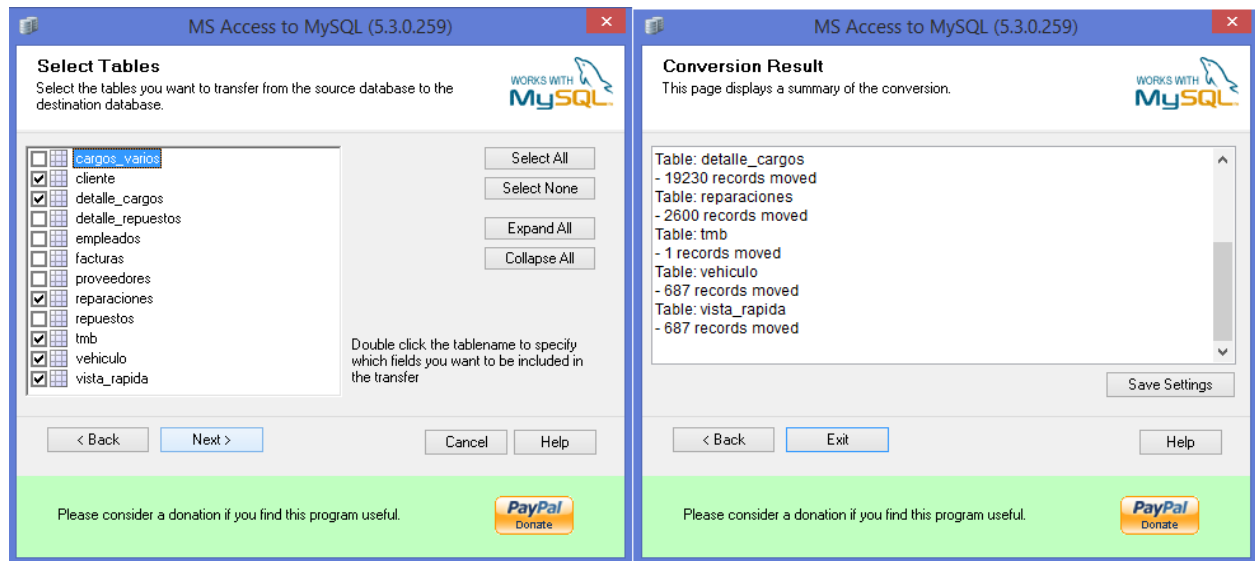


Ilustración 80. MS Access to MySQL

El siguiente paso es importar la Base de Datos a **phpMyAdmin**, la herramienta tiene un tiempo de importación limitado, que unido a un máximo de tamaño del archivo SQL, nos obliga a comprimir con **WinZip** el archivo para que la importación se lleve a cabo.

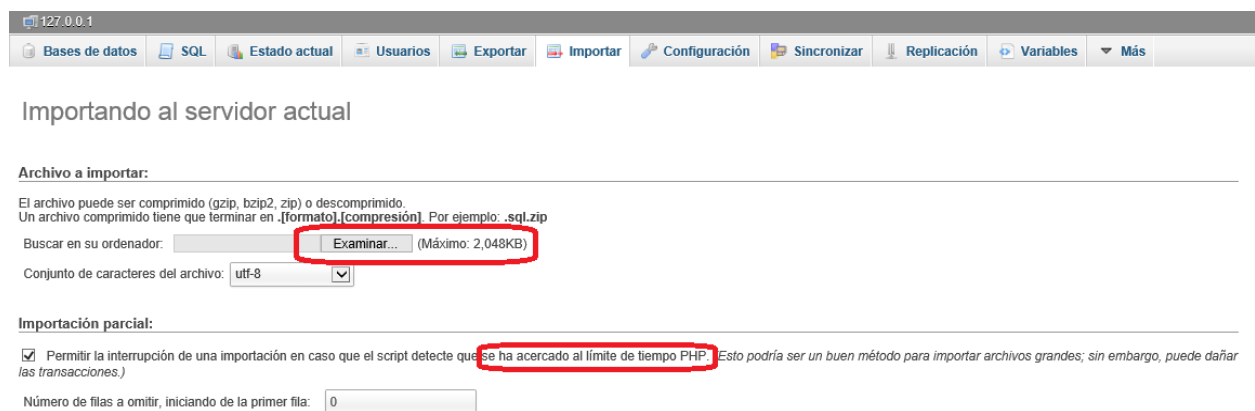


Ilustración 81. phpMyAdmin

Tras la importación, la Base de Datos resultante está en MyISAM y sin relaciones entre ellas. Utilizaremos un script para cambiar el método de almacenamiento y relacionarlas. El Script también optimizará la información y codificará las contraseñas.

Script de Ejecución:

```
CREATE TABLE `nuevo` (  
  
    `matricula` VARCHAR(8) NOT NULL,  
  
    `marca` VARCHAR(18),  
  
    `modelo` VARCHAR(20),  
  
    `matriculacion` DATETIME,  
  
    `tipo_motor` VARCHAR(15),  
  
    `chasis` VARCHAR(20),  
  
    `ncc` VARCHAR(10),  
  
    `pf` DECIMAL(19,4) DEFAULT 0,  
  
    `kw` DECIMAL(19,4) DEFAULT 0,  
  
    `prox_revision` INTEGER DEFAULT 0,  
  
    `fecha_reservada` DATETIME,  
  
    `prox_itv` DATETIME,  
  
    `distribucion` INTEGER DEFAULT 0,  
  
    `f_aire` VARCHAR(20),  
  
    `f_aceite` VARCHAR(20),  
  
    `f_habitaculo` VARCHAR(20),  
  
    `f_combustible` VARCHAR(20),  
  
    `aceite` VARCHAR(20),  
  
    `neumaticos` VARCHAR(150),
```

```
`intervalo_distribucion` VARCHAR(100),

`ref_distribucion` VARCHAR(20),

`cod_radio` VARCHAR(10),

`cod_llave` VARCHAR(10),

`aceitecc` VARCHAR(20),

`axone` VARCHAR(20),

`bujias` VARCHAR(20),

`dni` VARCHAR(9)

) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

insert into nuevo (matricula, marca, modelo, matriculacion, tipo_motor, chasis, ncc, pf, kw,
prox_revision, fecha_reservada, prox_itv, distribucion, f_aire, f_aceite, f_habitaculo,
f_combustible, aceite, neumaticos, intervalo_distribucion, ref_distribucion, cod_radio, cod_llave,
aceitecc, axone, bujias, dni)

select v.matricula, v.marca, v.modelo, v.matriculacion, v.tipo_motor, v.chasis, v.ncc, v.pf, v.kw,
r.prox_revision, r.fecha_reservada, r.prox_itv, r.distribucion, r.f_aire, r.f_aceite, r.f_habitaculo,
r.f_combustible, r.aceite, r.neumaticos, r.intervalo_distribucion, r.ref_distribucion, r.cod_radio,
r.cod_llave, r.aceitecc, r.axone, r.bujias, v.dni from vehiculo v, vista_rapida r

where v.matricula = r.matricula;

DROP TABLE `vehiculo`;

DROP TABLE `vista_rapida`;

RENAME TABLE `taller`.`nuevo` TO `taller`.`vehiculo` ;
```

```
ALTER TABLE `tmb` ADD `actualizacion` DATE;
```

```
UPDATE `tmb` SET `actualizacion`=NOW() WHERE `cif`='B81370041';
```

```
ALTER TABLE `cliente` ADD `contrasenia` VARCHAR(32);
```

```
UPDATE `cliente` SET `contrasenia`=MD5(`clave`);
```

```
ALTER TABLE `cliente` DROP `clave`;
```

```
ALTER TABLE `vehiculo` CHANGE `matriculacion` `matriculacion` DATE;
```

```
ALTER TABLE `vehiculo` CHANGE `fecha_reservada` `fecha_reservada` DATE;
```

```
ALTER TABLE `vehiculo` CHANGE `prox_itv` `prox_itv` DATE;
```

```
ALTER TABLE `reparaciones` CHANGE `fecha` `fecha` DATE;
```

```
ALTER TABLE `detalle_cargos` CHANGE `fecha` `fecha` DATE;
```

```
ALTER TABLE `tmb` ENGINE = InnoDB;
```

```
ALTER TABLE `cliente` ENGINE = InnoDB;
```

```
ALTER TABLE `vehiculo` ENGINE = InnoDB;
```

```
ALTER TABLE `reparaciones` ENGINE = InnoDB;
```

```
ALTER TABLE `detalle_cargos` ENGINE = InnoDB;
```

```
ALTER TABLE `tmb` ADD PRIMARY KEY ( `cif` );
```

```
ALTER TABLE `cliente` ADD PRIMARY KEY ( `dni` );
```

```
ALTER TABLE `vehiculo` ADD PRIMARY KEY ( `matricula` );
```

```
ALTER TABLE `vehiculo` MODIFY `dni` varchar(9) NOT NULL;
```

```
ALTER TABLE `reparaciones` ADD PRIMARY KEY ( `matricula`,`fecha` );
```

```
CREATE INDEX `index_reparaciones` ON `reparaciones` (`fecha`);
```

```
ALTER TABLE `detalle_cargos` ADD PRIMARY KEY ( `matricula`,`fecha`,`referencia` );
```

```
ALTER TABLE `vehiculo`
```

```
ADD FOREIGN KEY (`dni`) REFERENCES `cliente` (`dni`) ON DELETE CASCADE ON  
UPDATE CASCADE;
```

```
ALTER TABLE `reparaciones`
```

```
ADD FOREIGN KEY (`matricula`) REFERENCES `vehiculo` (`matricula`) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `detalle_cargos`
```

```
ADD FOREIGN KEY ( `matricula` ) REFERENCES `taller`.`reparaciones` ( `matricula` ) ON  
DELETE CASCADE ON UPDATE CASCADE ;
```

```
ALTER TABLE `detalle_cargos`
```

```
ADD FOREIGN KEY ( `fecha` ) REFERENCES `taller`.`reparaciones` ( `fecha` ) ON DELETE  
CASCADE ON UPDATE CASCADE ;
```


127.0.0.1 » taller											
Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas											
Tabla	Acción					Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar	
<input type="checkbox"/> cliente							578	InnoDB	utf8_general_ci	128 KB	-
<input type="checkbox"/> detalle_cargos							18,233	InnoDB	utf8_general_ci	3 MB	-
<input type="checkbox"/> reparaciones							2,466	InnoDB	utf8_general_ci	1.6 MB	-
<input type="checkbox"/> tmb							1	InnoDB	utf8_general_ci	16 KB	-
<input type="checkbox"/> vehiculo							674	InnoDB	utf8_general_ci	240 KB	-
5 tablas	Número de filas					21,952	InnoDB	latin1_swedish_ci	5 MB	0 B	

taller.tmb

- cif : varchar(20)
- nombre : varchar(80)
- direccion : varchar(50)
- provincia : varchar(50)
- localidad : varchar(50)
- cp : int(11)
- telf1 : varchar(9)
- telf2 : varchar(9)
- email : varchar(100)
- fax : varchar(9)
- n_registro : varchar(20)
- precio : decimal(19,4)
- iva : decimal(19,4)
- actualizacion : date

taller.cliente

- dni : varchar(9)
- nombre : varchar(18)
- ape1 : varchar(18)
- ape2 : varchar(18)
- direccion : varchar(50)
- localidad : varchar(30)
- provincia : varchar(20)
- cp : int(11)
- telf1 : varchar(9)
- telf2 : varchar(9)
- email : varchar(100)
- contrasenia : varchar(32)

taller.vehiculo

- matricula : varchar(8)
- marca : varchar(18)
- modelo : varchar(20)
- matriculacion : date
- tipo_motor : varchar(15)
- chasis : varchar(20)
- ncc : varchar(10)
- pf : decimal(19,4)
- kw : decimal(19,4)
- prox_revision : int(11)
- fecha_reservada : date
- prox_itv : date
- distribucion : int(11)
- f_aire : varchar(20)
- f_aceite : varchar(20)
- f_habitaculo : varchar(20)
- f_combustible : varchar(20)
- aceite : varchar(20)
- neumaticos : varchar(150)
- intervalo_distribucion : varchar(100)
- ref_distribucion : varchar(20)
- cod_radio : varchar(10)
- cod_llave : varchar(10)
- aceitecc : varchar(20)
- axone : varchar(20)
- bujias : varchar(20)
- dni : varchar(9)

taller.reparaciones

- matricula : varchar(8)
- fecha : date
- km : int(11)
- estado : varchar(10)
- ob_taller : longtext
- ob_cliente : longtext
- detalle : varchar(100)
- responsable : varchar(9)
- datos_tecnicos : varchar(80)
- numero : int(11)
- ano : int(11)

taller.detalle_cargos

- matricula : varchar(8)
- fecha : date
- referencia : int(11)
- tiempo_real : decimal(19,4)
- descripcion : varchar(100)
- grupo : int(11)
- subgrupo : int(11)

Ilustración 82. Base de Datos Optimizada

Con nuestra Base de Datos optimizada y relacionada, sólo queda exportar los datos cuando marque el Documento de Seguridad de la Empresa (**Anexo 5**), y almacenarla en el servidor del Proveedor de servicios Web.

6.3 Manual de Usuario Software de Gestión

A continuación se presenta el Manual de Usuario del **Software de Gestión TMB**, que realiza una descripción de las posibles acciones de la aplicación para la gestión del taller.



1.- El Programa de Escritorio

El Software de Escritorio TMB está programado para la administración informática de un taller de reparaciones de automóviles. Realiza la gestión de sus clientes, almacenando un histórico de las reparaciones de sus vehículos. Se encarga de la creación de las Órdenes de Trabajo y su posterior facturación, así como la gestión del Almacén y sus Proveedores.

2.- Inicio de la Aplicación

Al arrancar la Aplicación el sistema pide identificación al usuario para acceder al contenido. Las claves son proporcionadas por TMB.

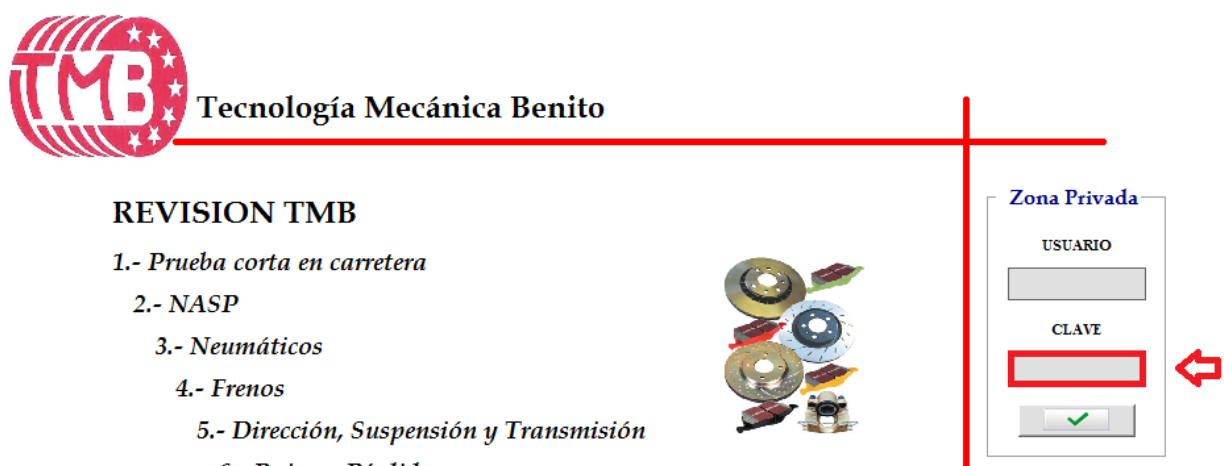


Ilustración 83. Inicio de la Aplicación

3.- Gestor de BBDD

Desde la pantalla principal del programa tenemos un acceso directo a la consola de administración de nuestra Base de Datos. En ella podemos **crear, modificar o eliminar** Clientes, Vehículos, Proveedores, Empleados y mantener al día nuestro Almacén.

Ilustración 84. Acceso Gestión BD

3.1.- Clientes

Es hora de empezar a “*inflar*” nuestra Base de Datos. Para crear un Cliente bastará con ir al menú **Cliente / Cliente nuevo (1)**. El sistema habilitará las cajas de texto y podremos rellenar los campos con información. Para guardar los datos debemos confirmar en el recuadro **Grabar Cliente (3)**, donde también tenemos la posibilidad de cancelar la operación. Las cajas de texto con fondo amarillo son campos obligatorios.

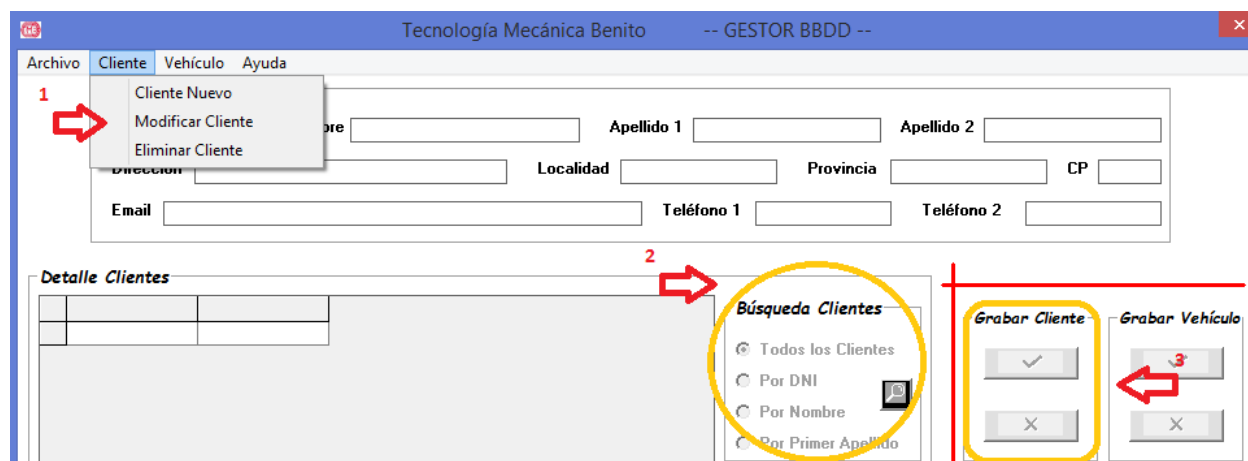


Ilustración 85. Gestor Clientes

Si por el contrario queremos modificar la información personal de un cliente, seguiremos la ruta **Cliente / Modificar Cliente (1)**. El buscador de clientes se habilitará para proporcionarnos un cuadro de **búsqueda por DNI, Nombre o Primer Apellido (2)**. Si el cliente es encontrado, las cajas de texto mostraran sus datos y podremos modificarlos. Esta acción puede confirmarse o cancelarse en el recuadro **Grabar Cliente (3)**.



Ilustración 86. Modificar Cliente

Eliminar un Cliente supone eliminar todos sus vehículos. “Pincharíamos” en **Cliente / Eliminar Cliente (1)** y tras seleccionar el cliente a borrar mediante **el Buscador (2)**, se nos pedirá **Confirmación del Borrado (3)**.



Ilustración 87. Eliminar Cliente

3.2.- Vehículos

Un taller de automóviles necesita vehículos. Para ingresar un vehículo nuevo en nuestra Base de Datos el dueño, nuestro cliente, ya debe estar registrado. Iríamos al menú **Vehículo / Vehículo nuevo** (1), buscaríamos el dueño del automóvil para relacionarlos a través del **Buscador de Clientes** (2) y rellenaríamos todas las cajas de texto pertenecientes a los datos técnicos del vehículo. Los campos en color amarillo son obligatorios.

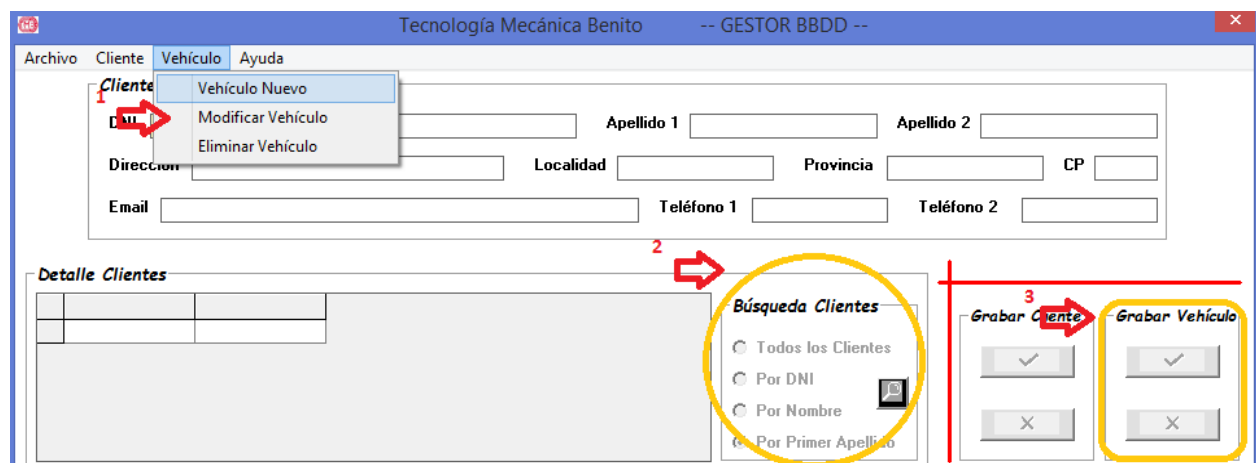


Ilustración 88. Gestor Vehículos

Detalle Clientes

Dni	Nombre	Apellido
53432441Z	ISMAEL	LOPEZ
53432661G	Mª CRISTINA	GARCIA
53432938M	MARTA	PARRA
53433005A	JOSE	GRANADOS
53433024E	ESTEFANIA	FERNANDEZ
53434040A	CARLOS	MORENO

Búsqueda Clientes

☐ Todos los Clientes
☐ Por DNI
☒ Por Nombre
☐ Por Primer Apellido

Nuevo Vehículo

Matrícula: Tipo Motor:
 Marca: Ncc:
 Modelo: PF:
 Matriculación: Kw:
 Chasis:

Vista Rápida

Prox. Revisión: Aceite Motor:
 Prox. Distribución: F. Aire:
 Ref. Distribución: F. Aceite:
 Neumáticos: F. Habitaculo:
 Inf. Axone: F. Combustible:
 Cod. Llave: Bujías:

Grabar Cliente **Grabar Vehículo**

Ilustración 89. Vehículo Nuevo

Tras ingresar la información, debemos confirmar la operación en el recuadro **Grabar Vehículo (3)**, donde también tenemos la posibilidad de cancelar la acción.

Si lo que queremos es modificar información relativa al automóvil, “*pincharemos*” en **Vehículo / Modificar vehículo (1)**, y un mensaje flotante nos pedirá la matrícula del vehículo a modificar. Si el vehículo existe en la Base de Datos, su información aparecerá en pantalla lista para ser actualizada. Toda acción debe confirmarse en el recuadro **Grabar Vehículo (3)**.

Modificar Vehículo

Introduzca Matrícula del Vehículo a Modificar:

Grabar Cliente **Grabar Vehículo**

Ilustración 90. Modificar Vehículo

Matrícula	M2637SV	Tipo Motor	G/CDZ	Prox. Revisión	224500	Aceite Motor	10W40
Marca	CITROEN	Ncc	4/954	Prox. Distribución	232000	F. Aire	C1468
Modelo	AX 1.0 I FIRST	PF	8,54	Ref. Distribución	KTB101	F. Aceite	W716/1
Matriculación	18/04/1995	Kw	37	Neumáticos	155/70 R13 75T	F. Habitaculo	NO HAY DATOS
Chasis	VS7ZALH0070LH2522			Inf. Axone	NO HAY DATOS	F. Combustible	NO HAY DATOS
				Cod. Llave	NO DATOS	Bujías	RC9YC
				Cod. Radio	NO DATOS	Aceite C/C	75W90
				Intervalo Distribución:	90000 KM O, 10 AÑOS.		

Ilustración 91. Información Vehículo a Modificar

De igual modo un vehículo puede ser suprimido de la Base de Datos, desapareciendo todas sus reparaciones. El cliente no se elimina ya que puede tener más vehículos. Seleccionaríamos **Vehículo / Eliminar Vehículo (1)**, apareciendo en pantalla un mensaje flotante solicitando la matricula del automóvil a borrar. De nuevo la acción requiere confirmación para llevarse a cabo.

Matrícula	M2537SV	Tipo Motor	G/CDZ
Marca	CITROEN	Ncc	4/954
Modelo	AX 1.0 I FIRST	PF	8,54
Matriculación	18/04/1995	Kw	37
Chasis	VS7ZALH0070LH2522		

Fecha Reservada	14/03/2016
ITV	17/03/2016

ATENCIÓN

ELIMINAR VEHICULO

Aceptar Cancelar

Cod. Radio

NO DATOS

Intervalo Distribución: 90000 KM O, 10 AÑOS.

Aceite Motor	10W40
F. Aire	C1468
F. Aceite	W716/1
F. Habitaculo	NO HAY DATOS
F. Combustible	NO HAY DATOS
Bujías	RC9YC
Aceite C/C	75W90

Ilustración 92. Eliminar vehículo

3.3.- Proveedores y Empleados

Dentro del Gestor de la Base de Datos, tenemos la posibilidad de mantener a los agentes que nos ayudarán a completar la reparación (Proveedores y Empleados). Accesibles desde el **Menú Archivo**.

Tecnología Mecánica Benito -- GESTOR BBDD --

Archivo Cliente Vehículo Ayuda

Almacén
Cargos Varios
Proveedores
Empleados
Datos TMB
Salir

Nombre Apellido 1 Apellido 2

Localidad Provincia CP

Teléfono 1 Teléfono 2

Detalle Clientes

Ilustración 93. Acceso Gestor BBDD II

Tecnología Mecánica Benito -- GESTOR BBDD --

Archivo Ayuda

Tecnología Mecánica Benito

CIF Nombre N° Registro IVA % Precio/Hora €

Dirección Localidad Provincia Cp

Teléfono 1 Teléfono 2 Fax Email

Proveedores ELIGE UNA OPCION

Empresa
(STD) STANDARD
ALFA (MEDIODIA)
ALMACEN
ANTONIO J. VA
ARDASA 2000, S
AUTOMATIC

CIF Nombre Email

Dirección Localidad Provincia Cp

Teléfono 1 Teléfono 2 Fax

Empleados ELIGE UNA OPCION

DNI	Nombre	Apellido	Puesto	Teléfono	Antigüedad
53463131E	LUIS	LOPEZ	AYUDANTE	925531431	01/09/2000

DNI Puesto

Nombre Apellido

Antigüedad Teléfono

Proveedores

Empleados

Ilustración 94. Gestor de BBDD II

Cada agente tiene su propia Consola de Mantenimiento. Si elegimos la opción **Proveedores**, en su sección podemos crear un nuevo proveedor, rellenando las cajas de texto que el sistema habilitará. Si por el contrario queremos **Modificar o Eliminar (2)** un proveedor, la **Tabla de Datos (1)** estará ordenada alfabéticamente para encontrar el registro.

Ilustración 95. Gestión de Proveedores

En la misma pantalla encontramos la consola de los **Empleados**, funcionando del mismo modo que los Proveedores. Con un nuevo empleado las cajas de texto se habilitan, y en las bajas y modificaciones hay que seleccionar el registro en la **tabla ordenada** (1). Las altas, bajas y modificaciones (2), tanto de Proveedores como Empleados, necesitan confirmación para **actualizar la Base de Datos** (3).

Ilustración 96. Gestión de Empleados

3.4.- Datos TMB

En la pantalla anterior, además de gestionar los proveedores y empleados, aparece información corporativa que también puede ser actualizada. **El Menú de botones** (1) permitirá modificar esta información de carácter empresarial.

Tecnología Mecánica Benito -- GESTOR BBDD --

Archivo Ayuda

Tecnología Mecánica Benito

CIF: Nombre: N° Registro: IVA %: Precio/Hora €:

Dirección: Localidad: Provincia: Cp: Modificar Datos Taller

Teléfono 1: Teléfono 2: Fax: Email:

Proveedores

Empresa	ELIGE UNA OPCION
(STD) STANDARD	
ALFA (MEDIODIA)	
ALMACEN	
ANTONIO J. VA	
ARDASA 2000,S	
AUTOMATIC	

CIF: Nombre: Email: Borrar Proveedor

Dirección: Localidad: Provincia: Modificar Proveedor

Teléfono 1: Teléfono 2: Fax: Cp: Nuevo Proveedor

Empleados

DNI	Nombre	Apellido	Puesto	Teléfono	Antigüedad	ELIGE UNA OPCION
53463131E	LUIS	LOPEZ	AYUDANTE	925531431	01/09/2000	

DNI: Puesto: Borrar Empleado

Nombre: Apellido: Modificar Empleado

Antigüedad: Teléfono: Nuevo Empleado

Proveedores

☐

☐

Empleados

☐

☐

Ilustración 97. Gestor de BBDD II

3.5.- Almacén y Cargos Varios

La Aplicación permite tanto la gestión del **Almacén** como de los **Conceptos de Reparación**. Los repuestos y los cargos varios darán forma a las Órdenes de Reparación y a la Facturación. Accedemos a su Pantalla de Mantenimiento a través de **Archivo/ Almacén**.

Tecnología Mecánica Benito -- GESTOR BBDD --

Archivo Cliente Vehículo Ayuda

Almacén

Cargos Varios

Proveedores

Empleados

Datos TMB

Salir

Nombre Apellido 1 Apellido 2

Localidad Provincia CP

Teléfono 1 Teléfono 2

Ilustración 98. Acceso Gestor Almacén / Cargos Varios

Esta nueva pantalla del Gestor de la Base de Datos funciona igual que las demás. Los Repuestos y los Cargos Varios se actualizan en la misma página pero en distintas Consolas de Mantenimiento.

En ambos casos cuando elegimos dar de **Alta un Registro (2)** las cajas de texto se habilitan para introducir la información. Para las **Modificaciones y Borrados disponemos de buscadores (1)** que facilitarán encontrar el registro a manipular. En los **Repuestos** tenemos la posibilidad de buscar por Descripción o por Referencia y Proveedor, en los **Cargos Varios** la búsqueda se realiza por descripción. Ambas “*tablas de datos*” están ordenadas alfabéticamente.

Todas las operaciones necesitan confirmación para actualizarse en la Base de Datos (3).

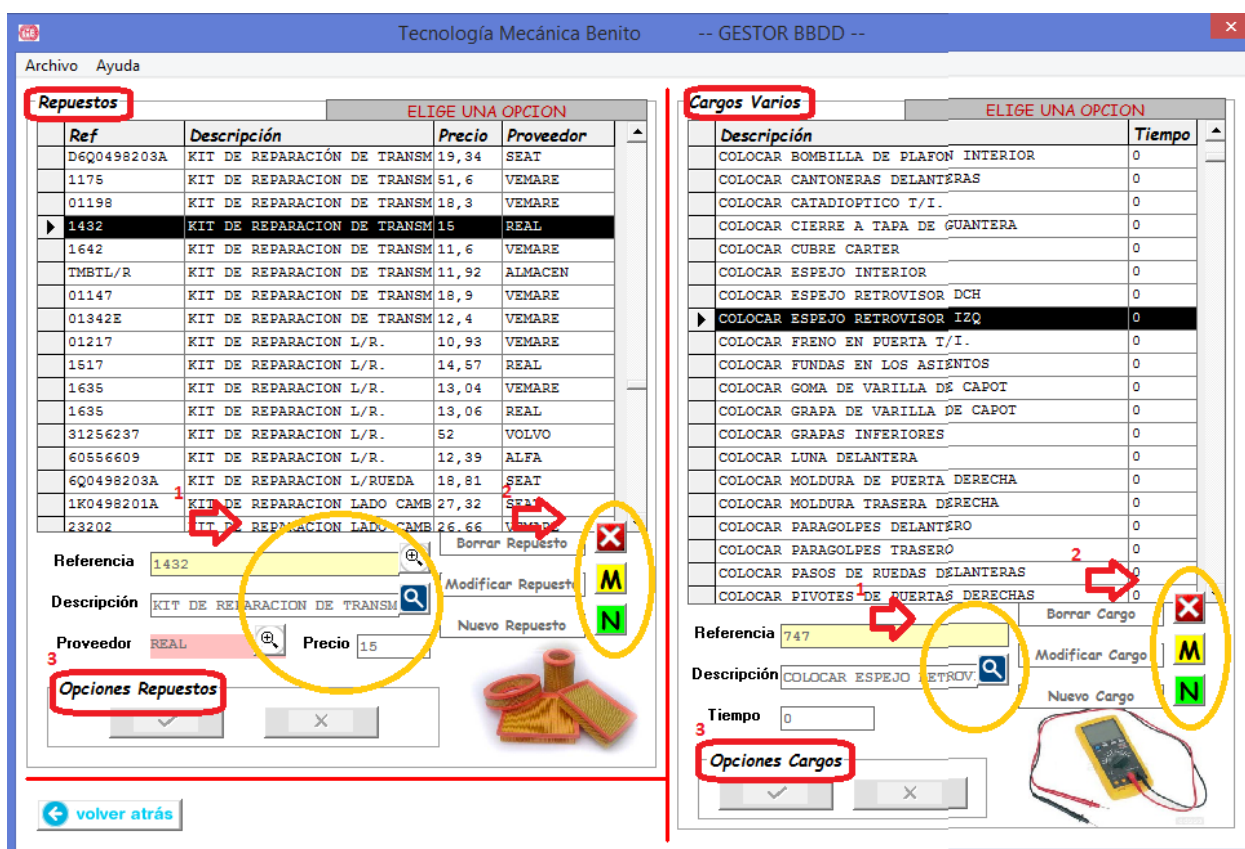


Ilustración 99. Gestor de Almacén / Cargos Varios

4.- Vista Rápida

Es la pantalla principal de la Aplicación y la que aporta más información general. De un sólo vistazo puedes ver la información personal del cliente, además de los datos técnicos y de mantenimiento de sus vehículos.

Ilustración 100. Pantalla Principal

Para encontrar el cliente a visualizar, existen diversos criterios de búsqueda que ayudarán a localizar el registro (**DNI, Matrícula, Nombre, Apellido 1**). Tras elegir la modalidad de búsqueda en **Archivo / Búsqueda Avanzada**, aparecerá en pantalla un dialogo de texto que solicitará el criterio de búsqueda. Si existiera más de un resultado, la aplicación mostrará una tabla con todos los resultados para elegir el adecuado.

Tecnología Mecánica Benito -- PANTALLA PRINCIPAL --

Archivo Gestor BBDD Ayuda

Reiniciar BBDD

Búsqueda Avanzada

- Búsqueda por DNI
- Búsqueda por Matrícula
- Búsqueda por Nombre
- Búsqueda por Apellido

Resguardo Depósito

Plástico Recogida

Salir

Vehículos

Matrícula: 8610GPB

Marca: PRINCEP

Apellido 1: FERNANDEZ, Apellido 2: ARROYO

VILLALUENGA DE L. Provincia: TOLEDO, CP: 45520

Teléfono 1: 627825640, Teléfono 2: 647410099

Fecha Reservada: 21/03/2016, Prox. Revisión: 107500

Prox. ITV: 20/09/2015, Prox. Distribución: 180000

Ilustración 101. Búsqueda Avanzada

Tecnología Mecánica Benito -- PANTALLA PRINCIPAL --

Archivo Gestor BBDD Ayuda

Cliente

DNI: 53426674C, Nombre: ANA, Apellido 1: VAQUERO, Apellido 2: TORRES

Dirección: C/LAGO BLANCO N°4 P7 3ªA, Localidad: PARLA, Provincia: MADRID, CP: 28980

Email: NO HAY DATOS, Teléfono 1: 662292399, Teléfono 2: NO DATOS

Vehículo

Búsqueda por Nombre: CARLOS

Matrícula	Dni	Nombre	Apellido 1	Apellido 2
53533023K		ALBERTO CARLOS	FERNANDEZ	BARQUILLA
75690697C		CARLOS	CABALLERO	GOMEZ
53434040A		CARLOS	MORENO	ESCRIBANO
53136098A		CARLOS	GARCIA	CHAMORRO
49013723K		CARLOS	LOPEZ	IGLESIAS
47282900Y		CARLOS	GARCIA	PEREZ
16077542J		CARLOS	BERMEJO	SILVANO
50747401V		CARLOS ALBERTO	MORENO	RUEDA
53431128N		JOSE CARLOS	GONZALEZ	RAMIREZ
50303435C		JUAN CARLOS	GOMEZ	ALVARO
70037059N		JUAN CARLOS	SERRANO	ROMASANTA

Fecha Reservada: 02/04/2012, Prox. Revisión: 135000

Prox. ITV: 07/04/2012, Prox. Distribución: 0

Búsqueda Cliente por Nombre

Introduzca Nombre a buscar: [CARLOS]

Aceptar Cancelar

Ilustración 102. Resultados de la Búsqueda

Un cliente puede tener varios vehículos. El sistema habilitará botones direccionales para desplazarnos por los distintos automóviles del cliente y ver su información técnica.

Chasis: VF34C9D6C9S131846

Selección Vehículo Cliente

Ir a Ficha de Vehículo Historial Vehículo

Aceite Cambio: 75W90, **Código Radio**: NO DATOS

Bujías: NO HAY DATOS, **Código Llave**: NO DATOS

Ref. Distribución: NO HAY DATOS

180000 KM. 0, 10 AÑOS (4/2019)

11:42:19 BÚSQUEDA POR DNI: 52981185H 19/09/2015

Ilustración 103. Selección de Vehículos

5.- Historial del Vehículo

Tras seleccionar nuestro vehículo en la pantalla principal, aparte de visualizar sus datos técnicos, tenemos acceso a su Libro de Mantenimiento.



Ilustración 104. Acceso Libro de Mantenimiento

En el Historial del Vehículo podremos ver todas las **reparaciones realizadas ordenadas por fecha (1)**. Cada reparación tiene sus Cargos de Varios y Repuestos propios, así como las observaciones del Cliente y del Taller.

El sistema ofrece la posibilidad de **Modificar o Eliminar (2)** una reparación seleccionada de la tabla de datos. Es en esta pantalla donde tenemos un acceso para realizar una **Orden de Reparación del vehículo seleccionado (2)**.

Tecnología Mecánica Benito -- HISTORIAL VEHÍCULO --

Archivo Ayuda

Fecha Reservada 21/03/2016

Próxima ITV 20/09/2015

Proxima Distribución 180000

Próxima Revisión 107500

Vehículo

Matrícula 8610GPB Tipo Motor D/9HZ

Marca PEUGEOT Ncc 4/1560

Modelo 308 SPORT HDI PF 11,45

Año 05/08/2009 Kw 80

Chasis VF34C9HZC9S131846

Detalle Reparaciones

Fecha	Km	Número	Año	Detalle
07/04/2015	92120	95	2015	NEUMATICOS
31/03/2015	92120	93	2015	RV/SF.1*

Fecha Reparación: 31/03/2015

Borrar Reparación

Detalle Repuestos

VEHICULO

Descripción	Referencia	Uds	Precio
ACEITE MOTOR	10W40	5	5,75
FILTRO DE ACEITE	HU716/2X	1	13,83
FILTRO DE AIRE	C3282	1	18
GRAPA DE S.DE BARILLA SUP.	792834	1	0,25
LIQUIDO DE FRENOS	DOT4	1	0
KIT DE FRENOS DELANTEROS	8132300	1	137,58
TIRANTA DE SUSPENSION	50-08519	1	24,65

DATOS TÉCNICOS REPARACIÓN 10W40

Observaciones Cliente

REVISION,VIERA EL VOLANTE ENTRE 100/110 KM./H., Y TIRA UN POCO A LA IZQUIERDA.

Modificar Reparación

Nueva Reparación

volver atrás

Ilustración 105. Libro de Mantenimiento

6.- Orden de Reparación

Las ordenes de reparación se construyen añadiendo **Conceptos de reparación y los Repuestos** necesarios para llevar a cabo esos trabajos. Utilizamos el botón de **Cambio de Vista** (1) para movernos de Conceptos de Reparación a Repuestos y viceversa.

Para ayudar a la creación de la Hoja de Trabajo, la aplicación ofrece **Información Técnica** (4) del vehículo y las **observaciones indicadas por el cliente y el Taller** (2). Todo ello mejorara la percepción del usuario a la hora de completar la orden.

En la misma pantalla, encontramos un **menú** (3) con acceso a facturación, y la posibilidad del imprimir la orden de reparación, el expediente del vehículo y el resguardo de depósito.

Tecnología Mecánica Benito -- HOJA DE TRABAJO VEHÍCULO --

Archivo Imprimir Plantillas Ayuda

Vehículo

Matrícula: 8610GPB DNI: 52981185H 31/03/2015

KM: 92120 Detalle: RV/SF.1*

Prox. Rev.: 107500 RSV: 21/03/2016 ITV: 20/09/2015

Nº Hoja Trabajo: 93 2015 HACIA RECAMBOS

Cargos Varios

Descripción
SUSTITUIR ROTULAS DE SUSPENSION DEL/INFERIORES
SUSTITUIR SBLOCK DE SOPORTE D/D.
SUSTITUIR SEGMENTO DNT. (TRINQUETE)
SUSTITUIR SEGMENTOS
SUSTITUIR SENSOR DE ALTURA
SUSTITUIR SENSOR DE ARBOL DE LEVAS
SUSTITUIR SENSOR DE CIGÜEÑAL
SUSTITUIR SENSOR DE FASE
SUSTITUIR SENSOR DE GIRO
SUSTITUIR SENSOR DE NIVEL DE ACEITE MOTOR
SUSTITUIR SENSOR DE NIVEL DE L.REFRIGERANTE
SUSTITUIR SENSOR DE NIVEL LIMPIAPARABRISAS.
SUSTITUIR SENSOR DE OCUPACION D/D.
SUSTITUIR SENSOR DE R.P.M.
SUSTITUIR SENSOR DE TEMPERATURA DE ADMISION

Tiempo: 0 Group: 1 1

Desc: SUSTITUIR SENSOR DE CIGÜEÑAL

Detalle Trabajos Realizados

Descripción	Tmp	G	SG
RV TMB	0	1	1
PREPARACION DEL TRABAJO	0	1	2
AUTODIAGNOSIS DE SISTEMAS Y RESET	0	1	3
SUSTITUIR GRAPA VARILLA DE CAPOT	0	1	4
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE	0	1	5
SUSTITUIR FILTRO DE AIRE	0	1	6
SUSTITUIR FILTRO DE HABITACULO	2,1	1	7
REPARACION DE FRENOS:	0	4	1
SUSTITUIR KIT DE FRENOS DELANTEROS	0	4	2
SUSTITUIR LIQUIDO DE FRENOS	0	4	3
SUSTITUIR TIRANTA DE SUSPENSION D/D.	1,8	5	1

REPARACION DE FRENOS: 4 1 Borrar

DATOS TÉCNICOS REPARACIÓN 10W40

Observaciones Taller

RUIDO POR ZONA DE DISTRIBUCION.
M,S/I:P1461

Grabar Reparación

volver atrás

Ilustración 106. Creación Hoja de Trabajo

Para añadir Repuestos a la Hoja de Trabajo, el sistema ofrece un **histórico de referencias (2)** para facilitar la labor del usuario. También puedes encontrar la referencia mediante sus buscadores o crear el repuesto directamente (1). Cada repuesto que utilicemos en la orden de reparación es único, tiene su referencia, precio y proveedor.

Tecnología Mecánica Benito -- HOJA DE TRABAJO VEHÍCULO --

Archivo Imprimir Plantillas Ayuda

Vehículo
 Matricula 8610GPB DNI 52981185H 31/03/2015
 KM 92120 Detalle RV/SF.1*
 Prox. Rev. 107500 RSV 21/03/2016 ITV 20/09/2015

N° Hoja Trabajo 93 2015 HACIA CARGOS VARIOS

Repuestos

Rf	Descripción	Proveedor
21103.00	JUEGO DE PASTILLAS TRASERAS	DEMARE
2263.51	JUEGO DE PASTILLAS TRASERAS	DEMARE
JABON GEL	10 KG	BRUDISA
6351AR	3ª LUZ DE FRENO	PEUGEOT
13433108	ABRAZADERA	ALFA
250245	ABRAZADERA	DEMARE
N90654001	ABRAZADERA	ARDASA 2000
16283J	ABRAZADERA EGR	PEUGEOT
75W80	ACEITE C/ CAMBIOS	ALMACEN
9730A5 PEUGEOT	ACEITE C/C	PEUGEOT
75W90	ACEITE C/CAMBIOS	BRUDISA
DIII	ACEITE CAJA DE CAMBIOS	DEMARE
1940182	ACEITE DE CAJA DE CAMBIOS	OPEL
AD ATF III DEX RO	ACEITE DE CAJA DE CAMBIOS	DEMARE

Ref N90654001 Proveedor ARDASA 2000
 Desc ABRAZADERA Precio 2.1 Uds

Añadir Repuesto

Detalle Repuestos

Ref	Descripción	Uds	Precio
10W40	ACEITE MOTOR	5	5,75
HU716/2X	FILTRO DE ACEITE	1	13,83
C3282	FILTRO DE AIRE	1	18
792834	GRAPA DE S.DE BARILLA SUP.	1	0,25
DOT4	LIQUIDO DE FRENOS	1	0
8132300	KIT DE FRENOS DELANTEROS	1	137,58
50-08519	TIRANTA DE SUSPENSION	1	24,65

Borrar Registro

Vista Rápida

Actualizar Datos

F. Aceite 10W40 F. Combustible NO HAY DATOS
 F. Aire HU716/2X Bujías NO HAY DATOS
 F. Habitaculo CU2940 Aceite Cambio 75W90
 Neumáticos 205/55 R16 91V
 Prox.Distribución 180000
 Distribución: 180000 KM. O,10 AÑOS (4/2019)

Ilustración 107. Repuestos de la Orden de Reparación

Los conceptos de reparación funcionan de forma parecida. El programa presenta un buscador para encontrar el concepto deseado (1), si no existe, ofrece la posibilidad de crearlo. El concepto de la orden es único para ella, por lo que tiene su propio tiempo de reparación.

SUSTITUIR SOPORTE CENTRAL DE GRUPO
 SUSTITUIR SOPORTE COMPRESOR DE A/C.
 SUSTITUIR SOPORTE DE CAJA DE CAMBIOS
 SUSTITUIR SOPORTE DE ESCAPE
 SUSTITUIR SOPORTE DE PINZA D/12
 SUSTITUIR SOPORTE DE SILENCIOSO TRASERO
 SUSTITUIR SOPORTE DERECHO DE MOTOR
 SUSTITUIR SOPORTE INF/DCH DE MOTOR
 SUSTITUIR SOPORTE INFERIOR DE MOTOR

tiempo 2 Group 1 1

Desc SUSTITUIR SOPORTE DE ESCAPE

REPARACION DE FRENOS: 4 1 Borrar

DATOS TÉCNICOS REPARACIÓN 10W40

Observaciones Taller

RUIDO POR ZONA DE DISTRIBUCION.
 M,S/I:P1461

Ilustración 108. Conceptos de Reparación

7.- Facturación

Cuando una orden de reparación se cierra, el trabajo ya se ha realizado, el sistema está preparado para facturar.

Tecnología Mecánica Benito -- FACTURACIÓN --

Archivo Ayuda

Vehículo

Matrícula 8610GPB DNI 52981185H RSV 21/03/2016 ITV 20/09/2015

KM 92120 Prox. Revisión 107500 Detalle RV/SF.1* Fecha 19/09/2015

☒ REVISION TMB
☒ PREPARACION DEL TRABAJO
☒ AUTODIAGNOSIS Y RESET

Detalle Repuestos

Ref	Descripción	Uds	Precio	Dto	Importe
10W40	ACEITE MOTOR	5	5,75	0	5,75
HU716/2X	FILTRO DE ACEITE	1	13,83	0	13,83
C3282	FILTRO DE AIRE	1	18	0	18
792834	GRAPA DE S.DE BARILLA SU	1	0,25	0	0,25
DOT4	LIQUIDO DE FRENOS	1	0	0	0
8132300	KIT DE FRENOS DELANTEROS	1	137,58	0	137,58
50-08519	TIRANTA DE SUSPENSION	1	24,65	0	24,65

GRAPA DE S.DE BARILLA SUP. 0.25 0 0,25

TOTAL MATERIALES 223,06

Detalle Trabajos Realizados

Descripción	Tmp	G	SG
RV TMB	0	1	1
PREPARACIÓN DEL TRABAJO	0	1	2
AUTODIAGNOSIS DE SISTEMAS Y RESET	0	1	3
SUSTITUIR GRAPA VARILLA DE CAPOT	0	1	4
CAMBIO DE ACEITE MOTOR Y FILTRO DE ACEITE	0	1	5
SUSTITUIR FILTRO DE AIRE	0	1	6
SUSTITUIR FILTRO DE HABITACULO	2,1	1	7
REPARACION DE FRENOS:	0	4	1
SUSTITUIR KIT DE FRENOS DELANTEROS	0	4	2
SUSTITUIR LIQUIDO DE FRENOS	0	4	3
SUSTITUIR TIRANTA DE SUSPENSION D/D.	1,8	5	1

SUSTITUIR TIRANTA DE SUSPENSION 1.8 5 1

TIEMPOS 3,9 **TOTAL MANO DE OBRA** 144,3

Datos Factura

Precio/Hora € 37 IVA % 21 Total IVA € 77,15

TOTAL REPARACION € 367,36 TOTAL FACTURA € 444,51

Observaciones Taller

RUIDO POR ZONA DE DISTRIBUCION.
M,S/I:P1461

[volver atrás](#)

Número de Factura 98 2015

Ilustración 109. Facturación

En la pantalla aparecerán todos los conceptos de reparación con sus tiempos, y los repuestos con sus precios. Todos estos datos pueden modificarse, agrupar los conceptos y crear descuentos (1).

La pantalla ofrece los valores generales de la factura, que pueden ir *refrescándose* (2) mediante el botón “calculadora” (2), a medida que modificamos la información. En la factura también aparecerán las observaciones del Taller, por lo que pueden editarse.

6.4 Manual de Usuario Libro de Mantenimiento online

A continuación se presenta el manual de usuario registrado del Portal Web, que realiza una descripción de las posibles acciones en la zona privada del cliente.



1.- El Portal Web

La zona privada de la Web del Taller, ofrecerá al **usuario registrado** la posibilidad de consultar el histórico de las reparaciones de sus vehículos. El cliente encontrará información de contacto y noticias del Mundo del Motor, también visualizará un menú de ofertas personalizadas de las que puede beneficiarse.

2.- Acceso Zona Privada

El cliente debe identificarse para entrar en su zona privada, el enlace se encontrará en la parte superior de la Web. Las claves serán proporcionadas por TMB.



Ilustración 110. Portada Portal Web

The image shows the login page of the website. It features a central white box on a light gray background. Inside the box, the title 'Control Acceso' is at the top. Below it are two input fields: 'Nombre De Usuario' with the text 'Alfredo' and 'Contraseña' with a masked password represented by dots. An orange 'Entrar' button is located at the bottom right of the input fields. Below the white box, there is a dark gray banner with white text that reads: 'TMB le ha proporcionado una clave secreta para acceder a su zona privada. Por razones de seguridad, si ha olvidado su clave, o la ha extraviado, póngase en contacto con el taller.'

El sistema verificará si el cliente tiene permisos de acceso, en caso negativo denegará el acceso y volverá a solicitar las claves.

3.- Pantalla Principal

En la pantalla principal el usuario visualizará todos los **vehículos de su propiedad (2)**. Mediante un enlace accederá a las reparaciones de cada uno de ellos. El cliente también tendrá acceso en esta pantalla a las **noticias y ofertas personalizadas (3)**.

Un **menú de navegación (1)** ayudará al cliente a moverse entre pantallas, destacando en la parte superior, la fecha de actualización de su información y el **enlace de desconexión (4)**, para abandonar de forma segura su sesión.

El cliente tendrá en todo momento la posibilidad de contactar con la empresa, mediante un enlace permanente en la parte superior de su zona privada.



Ilustración 111. Pantalla Principal



Ilustración 112. Ampliación Pantalla Principal

4.- Selección de Reparaciones

El usuario tras seleccionar el vehículo a consultar, visualizará un histórico con todas las **reparaciones que se han realizado al automóvil (2)**. La tabla de datos estará ordenada por fecha de reparación y tendrá acceso al desglose de cada una de ellas.

En esta pantalla el cliente también puede consultar **fechas de mantenimiento importantes (1)**, como la Fecha de ITV y la Fecha Reservada. La Próxima Revisión también es visible.

Inicio Aviso Legal Contacto

área privada
selecciona la reparación



1 →

Matrícula	4293DSF
Próxima Revisión	235000 Km
Próxima ITV	28/11/2015
Fecha Reservada	06/07/2015

2 →

FECHA	KIL6METROS	DETALLE REPARACIÓN
14/12/2014	231000	NEUMATICOS
08/07/2014	221350	RV/SF;
15/11/2013	205445	GOLPE
03/07/2013	199225	RV/SF;
10/05/2013	194937	RETROVISOR; A/AC;
15/10/2012	185814	LUZ DIRECCION;RV/SF.
31/05/2012	176000	CARGA A/AC;

Ilustración 113. Reparaciones del Vehículo

5.- Resumen de Reparaciones

Los conceptos de reparación indican el trabajo realizado al vehículo. El cliente visualizará la información en **la tabla de datos (2)**, ordenada por grupos de reparación. También aparecen en pantalla la **Fecha de la Reparación y los Kilómetros que tenía el vehículo (1)**.



Ilustración 114. Detalle de Reparación

6.5 Backup Documento de Seguridad (Migración BBDD)

La Migración de la Base de Datos del Programa de Escritorio (Access), al Almacén de Datos con el que trabajará el Portal Web (**MySQL**), se realizará durante el **Backup** que hace la empresa siguiendo su Documento de Seguridad.

A continuación se adjunta los artículos del Documento de Seguridad referentes a este respaldo, tanto el momento y periodicidad de la recuperación, como la descripción de los ficheros a copiar.

DOCUMENTO

de Seguridad de Datos

TECNOLOGIA MECANICA BENITO S.L.

V 1.0 29/07/2015

Índice

DOCUMENTO DE SEGURIDAD TECNOLOGIA MECANICA BENITO S.L.



ÁMBITO DE APLICACIÓN DEL DOCUMENTO



MEDIDAS, NORMAS, PROCEDIMIENTOS, REGLAS Y ESTÁNDARES
ENCAMINADOS A GARANTIZAR LOS NIVELES DE SEGURIDAD EXIGIDOS EN ESTE
DOCUMENTO

.....

Copias de respaldo y recuperación

Se realizarán copias de respaldo, salvo que no se hubiese producido ninguna actualización de los datos, con la siguiente periodicidad:

1 vez por semana (**Viernes al cierre**) se realizará una copia total de los datos.

Los procedimientos establecidos para las copias de respaldo y para su recuperación garantizarán su reconstrucción en el estado en que se encontraban al tiempo de producirse la pérdida o destrucción.

El Responsable del Fichero verificará semestralmente los procedimientos de copias de respaldo y recuperación de los datos.

Las pruebas anteriores a la implantación o modificación de sistemas de información se realizarán con datos reales previa copia de seguridad, y garantizando el nivel correspondiente al tratamiento realizado.

En el Anexo I se detallan los procedimientos de copia y recuperación de respaldo para cada fichero.

.....



INFORMACIÓN Y OBLIGACIONES DEL PERSONAL



PROCEDIMIENTOS DE NOTIFICACIÓN, GESTIÓN Y RESPUESTA ANTE LAS
INCIDENCIAS



PROCEDIMIENTOS DE REVISIÓN

ANEXO I – DESCRIPCIÓN DE FICHEROS

Actualizado a: 29 de Julio de 2015.

- Nombre del fichero o tratamiento: **Clientes y Proveedores.**
- Unidad/es con acceso al fichero o tratamiento: **Área de Administración**
 - Identificador y nombre del fichero en el Registro General de Protección de Datos de la Agencia Española de Protección de Datos:
- Identificador:
- Nombre: Clientes y Proveedores
- Descripción: Gestión de Clientes y Proveedores contable y administrativa
 - Nivel de medidas de seguridad a adoptar: **Básico.**
 - Administrador: **Responsable de Seguridad**
 - Leyes o regulaciones aplicables que afectan al fichero o tratamiento –
 - Código Tipo Aplicable:
–
 - Estructura del fichero principal: **Dni/Cif, Nombre, Apellidos, Dirección, Teléfonos tanto de los clientes como de los proveedores. Matrícula y Datos técnicos necesarios de los vehículos de los clientes para llevar a cabo la reparación.**
 - Información sobre el fichero o tratamiento
 - Finalidad y usos previstos.
Gestión interna de relaciones laborales
Taller/ Cliente, Taller/Proveedor
 - Personas o colectivos sobre los que se pretenda obtener o que resulten obligados a suministrar los datos personales, y procedencia de los datos: **Los datos son proporcionados directamente por los interesados (Clientes y Proveedores), siendo informados todos ellos del trato que tendrá la información cedida.**
 - Procedimiento de recogida: **Resguardos de depósito, tarjetas de visita, o datos proporcionados directamente por el interesado.**

- Cesiones previstas:
 -
- Transferencias Internacionales:
 -
- Sistema de tratamiento: **Mixto.**
- Servicio o Unidad ante el que puedan ejercitarse los derechos de acceso, rectificación, cancelación y oposición:
Tecnología Mecánica Benito SL
Calle San Sebastián 2; 28982, Parla (Madrid)
916981288
- Descripción detallada de las copias de respaldo y de los procedimientos de recuperación: **Copias de seguridad realizadas cada semana (Viernes al cierre del Taller)**
- Información sobre conexión con otros sistemas:
 -
- Funciones del personal con acceso a los datos personales:

Benito López Vaquero,

Usuario de fichero, Gestión administrativa del sistema de datos.

M^a Carmen Luque Murez,

Usuario de fichero, Gestión administrativa del sistema de datos.

Luis Marino López Luque,

***Administrador, Administrador del sistema de datos
y Responsable de Seguridad.***

- Descripción de los procedimientos de control de acceso e identificación:

Cada usuario con acceso autorizado tendrá un par de información individual para acceder al sistema: Contraseña y Nombre de Usuario

- Relación actualizada de usuarios con acceso autorizado:

Benito López Vaquero,

blv00001, Usuario de fichero

Fecha Alta / Baja 29/07/2015 // ...

M^a Carmen Luque Murez,

mlm00002, Usuario de fichero

Fecha Alta / Baja 29/07/2015 // ...

Luis Marino López Luque, lll00003

lll00003, Administrador

Fecha Alta / Baja 29/07/2015 // ...

ANEXO II – NOMBRAMIENTOS

ANEXO III – AUTORIZACIONES DE SALIDA O RECUPERACIÓN DE DATOS

ANEXO IV – DELEGACIÓN DE AUTORIZACIONES

ANEXO V – INVENTARIO DE SOPORTES

ANEXO VI – REGISTRO DE INCIDENCIAS

ANEXO VII – ENCARGADOS DE TRATAMIENTO

ANEXO VIII – REGISTRO DE ENTRADA Y SALIDA DE SOPORTES

ANEXO IX – MEDIDAS ALTERNATIVAS

6.6 Glosario

ADO	Objeto para acceso a datos
API	Interface de programación de Aplicaciones
BATCH	Archivo .bat que contiene una secuencia de órdenes para ser ejecutadas
BBDD	Bases de Datos
COM	Modelo de Objetos orientado a componentes
CSS3	Especificación de Hoja de Estilos en Cascada
DAO	Objeto para acceso a datos
GUI	Interfaz Gráfica de Usuario
HTML5	Quinta revisión del lenguaje de marcas HTML
IDE	Ambiente de Desarrollo Integrado
ITV	Inspección Técnica de Vehículos
MD5	Algoritmo de reducción criptográfico
MYSQL	Sistema Gestor de BBDD
ODBC	Conectividad Abierta de BBBDD
OLE DB	Enlace e incrustación de objetos para bases de datos

PFC	Proyecto Fin de Carrera
PHP	Lenguaje de Programación con código del lado del servidor
PLUGIN	Una aplicación complemento
PYME	Pequeña y Mediana Empresa
SVG	Gráficos Vectoriales Redimensionables
TMB	Tecnología Mecánica Benito
W3C	Consortio internacional que produce recomendaciones para la World Wide Web
XAMPP	Servidor independiente de plataforma (Apache, MySQL, PHP, Perl.)
XSS	Referente a los Ataques Xss. (Inyección de Javascript)